ROBUST AND ADAPTIVE RECEIVER DESIGN FOR WIRELESS COMMUNICATION SYSTEMS

Zheng Li

BE, ME Electrical Engineering

Shanghai Jiao Tong University, China

April 2000

A thesis submitted for the degree of Master of Engineering of the Australian National University

Department of Engineering

Faculty of Engineering and Information Technology

The Australian National University

Declaration

The work contained in this thesis, except where explicitly stated, is original research performed by the author. This work has not been submitted for a degree at any other university or institution.

These studies were conducted under supervision of Dr. Lei Wei, Dr. Matthew James, and Dr. Rodney Kennedy. Dr. Wei and Dr. James are with Department of Engineering, Faculty of Engineering and Information Technology, The Australian National University. Dr. Kennedy is with Telecommunication Group, Research School of Information and System Engineering, The Australian National University.

Zheng Li,

Canberra, November 1999

Acknowledgement

The author is fortunate to have Dr. Lei Wei, Dr. Matthew James and Dr. Rodney Kennedy as my supervisor and advisors.

Dr. Wei introduced me into the field of telecommunication engineering in the beginning and spent great time explaining to me the basics of multiuser detection, Viterbi algorithm and error control coding. Most of my research work comes out of our frequent discussion and joint effort on numerous topics. I owe great thanks to Dr. Wei for his supervision.

Dr. James introduced me into the area of robust signal processing and contributed tremendously to the two robust related publication. I'm glad I can help him with the lab work of ENGN3001 Digital Design. Dr. James also gave me lots of support in my career planning.

Dr. Kennedy gave me great support during my mid-term review and final thesis writing. And he is also kind enough to give me general advice on research and career planning.

I'd also like to thank Dr. Ian Petersen for his kind comments on my two robust related publication. Dr. Brian Hart gave me a valuable tutorial on fading channel. Dr. Predrag Rapajic gave me several interesting talks on adaptive filter design and practical communication channels.

The author want to thank Bob Edwards for his help resolving computing issues and the ladies of our department: Sue, Josephine and Sharon for administrative support.

It's been great to work with Mr. Yung Chung (Michael) Wei, Mr. Qi Wang and Mr. Lei Qiu. It's a balanced group with people of different talents, I learn a lot from each and every of you almost every day. I'll remember this experience. I'd also like to thank all the students I know in this department: Endra Julianto for his LaTeX tips and his wife's dinner, William Paul Malcolm (WPM) for his LaTeX tips and "coffee chats", Junmei, Aoshuang, Alma, Bernie, Paul Stacy, Paul Compston, Zisi Liu, Michelle, Narong(Tom) and Holger for the interesting conversations, all the basketball and soccer players for the fun.

Last but not least, my sincere gratitude to my family, who gave me incredible moral support all through the years, which make all the work meaningful.

Abstract

This thesis addresses the adaptive and robust receiver design task for wireless communication systems. We compare different combinations of adaptive schemes, matched or whitened matched filter front-end and sequence detector back-end, in order to adaptively estimate system parameters for Intersymbol Interference (ISI) channel and multiuser detection in Code Division Multiple Access (CDMA) system. We propose a novel robust detecting kernel to deal with channels populated by a mixture of noise with different probability density function (PDF). This robust kernel is applied to Viterbi algorithm (VA) and several *a posteriori* probability (APP) algorithms for Turbo code and Low-Density Parity-Check (LDPC) code.

We start with a brief discussion on several powerful decoding algorithms including Viterbi algorithm (M-algorithm) and APP algorithms. When we have perfect knowledge of the channel parameters, receivers designed according to these parameters could achieve the optimal theoretical performance. However, if we only know partial information about the channel, or if the channel is continuously changing in an unpredictable way, the receivers designed upon imperfect knowledge often perform far worse than the optimal one. This could easily undermine the precious gain hard-earned by using various powerful channel coding and decoding methods. And it is particularly true in the wireless channel.

The adaptive scheme continuously estimates the system parameters and trace them in the receiver accordingly. The focus on detector back-end is the maximum likelihood detector Viterbi algorithm and its reduced-complexity cousin M-algorithm, both used for sequence detection in ISI channel and multiuser detection in direct sequence CDMA system. In the receiver filter front-end, both matched

filter and whitened matched filter structures are explored. Using minimum mean square error estimation technique, we propose a joint adaptive method and a channel estimation method. All of the above front-ends, back-ends and adaptive schemes inter-weave a rich set of combination, which is fully studied and compared for their relative merits and disadvantages. This results in some important observation: the whitened matched filter couples better with M-algorithm; the joint adaptation method is simpler while the channel estimation method generally performs slightly better. In a word, these results will be valuable in helping design adaptive receivers.

The robust scheme uses a minimax decoder kernel to minimize the maximum error probability among a set of noise PDF. The "minimax" concept try to optimize for the worst possible case. We defined "Likelihood Separation Metric" (LSM) to evaluate the relative difficulty of correctly detecting the transmitted symbols in the presence of various noise PDF. Then we calculate this metric for each noise PDF at each time interval and select the decoder matched to the worst noise (smallest metric). The robust decoders always performs better than the worst mismatched decoder and very close to the optimal decoder. This robust kernel is readily implementable for a wide spectrum of decoding algorithms, such as Viterbi algorithm for convolutional code, MAP algorithm for Turbo code and *a posteriori* probability algorithms for low-density parity check code and general graph based codes. There is no or little computational overhead for adding the robust scheme on top of the traditional decoder if branch metric lookup table is computed offline. In a word, our robust scheme is both simple and effective, and can be used complimentarily with other noise estimation methods proposed in recent robust decoding literature.

Both of our adaptive and robust receivers prove to be valuable for their designated design purpose. Our comprehensive numerical simulation and analytical results strongly support these conclusions. At the end of this thesis, we will also propose several new directions in extending the above work.

Glossary

Abbreviations

AMPS	Advanced Mobile Phone System
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
CDMA	Code Division Multiple Access
dB	decibel
DFE	Decision-feedback Equalizer
DS-CDMA	Direct-Sequence Code Division Multiple Access
FDMA	Frequency Division Multiple Access
FIR	Finite Impulse Response
GSM	Global System for Mobile Communications
IIR	Infinite Impulse Response
IMT-2000	International Mobile Telecommunication for 2000
ISI	Inter-Symbol Interference

LDPC	Low-Density Parity Check Code
LSM	Likelihood Separation Metric
MA	M-algorithm
MAP	Maximum *a posteriori* probability
MF	Matched Filter
MLSD	Maximum Likelihood Sequence Detection
MMSE	Minimum Mean Square Error
MUD	Multiuser Detection
MUI	Multiuser Interference
PDF	Probability Density Function
SIR	Signal to Interference Ratio
SNR	Signal to Noise Ratio
TDMA	Time Division Multiple Access
VA	Viterbi Algorithm
WMF	Whitened Matched Filter

Contents

List of Figures

Chapter 1

Introduction

Vigourous writing is concise. ... This requires not that the writer make all his sentences short, or that he avoid all detail and treat his subject only in outline, but that every word tell.

- William Strunk Jr., "The Elements of Style"

1.1 Background

1.1.1 The need for speed

In recent years, there is increasing hunger for more speed out of the communication systems, thanks to the bandwidth-thirsty applications such as streamed audio/video and graphic intensive world wide web traffic. It is predicted that in a global point of view, data related traffic will soon surpass voice related traffic in the telecommunication networks in the very near future. Actually in some countries, this pattern of traffic is already a matter of reality [35] [63].

To meet the challenge of such dramatic change in the traffic volume and pattern, the old faithful voice-oriented telecommunication systems have to be overhauled [59] [61] [49]. Today the backbone is pretty much upgraded to ultra high-capacity (hundreds of gigabit to terabits per second) and exceptionally high quality (high signal to noise ratio, low attenuation) fibre optical transmission, thanks to the dense wavelength division multiplexing technology and light amplifier technology based on Erbium-doped fibre [64]. On the business side, due to the extensive de-regulation waves sweeping through all the continents, there have never been more companies in history than today that are beavering around wiring up the globe with fibres [23] [62].

With pressure on the backbone relatively eased, the bottlenecks now appear to fall in the user local loop, a telecommunication term which means the connection between the subscriber and the end office of service provider [17] [18] [36]. In today's communications landscape, the local loop will usually take the form of, among many others, the twisted copper wire telephone line, the coaxial cable from your pay TV company or the wireless radio link between a mobile phone and a base-station. These local loops are heavily polluted with environmental and man-made noise, crowded with lots of inter-user and inter-cell interference, shadowed and faded. In a word, they are telecommunication engineer's nightmare.

1.1.2 Powerful algorithms

To squeeze more speed out of such "bad" channels while maintaining high quality transmission, powerful algorithms are needed to replace traditional algorithms.

In communication theory, it has long been known that a sequence based maximum likelihood detector (such as the Viterbi algorithm) or *a posterior* probability detector (such as the one used in Turbo decoder) are optimal. But the computational complexity of such algorithms are prohibitive when the receiver gets more complicated. In the past, it is the feasibility that prevented the successful deployment of these powerful algorithms in commercial communication system, although they have been extensively used in military and deep space communication systems since their inception.

In recent years, due to the significant breakthrough in the computing and microchip technology, we have so much computing power at such a low cost, low power consumption and small physical profile that we can implement some really sophisticated algorithms in the communication device- not only at the base-station or central switch level, but also at handset level. So there is refreshed enthusiasm in both academia and industry to further explore the application of those optimal algorithms.

There are also a wealth of reduced complexity algorithms based on the above optimal algorithms, which show a very promising compromise between complexity and performance. They generally have performance much better than traditional algorithms, and in most cases close to the significantly more complicated optimal one. They have increased computational demand than traditional algorithms, but not to the degree typically associated with most optimal algorithms.

So to our special interests are the following powerful algorithms (with more details discussed in their corresponding chapters):

1. The Viterbi algorithm and reduced-complexity M-algorithm in inter-symbol interference Channels; (Chapter 3)

2. The Viterbi algorithm and reduced-complexity M-algorithm in multiuser detection for direct sequence Code Division Multiple Access systems; (Chapter 4)

3. Viterbi algorithm for convolutional code decoding; (Chapter 5)

4. Maximum *a posteriori* probability (MAP) decoding algorithm for Turbo code; (Chapter 6)

5. *A posteriori* probability decoding algorithm for Low Density Parity Check codes and general graph based codes. (Chapter 6)

1.2 The Uncertain Elements and the Two White Knights to the Rescue

Unfortunately, these powerful algorithms are prone to performance degradation when there are uncertainties in the communication system, which will seriously jeopardize their usability in practice. So we resort to the traditional wisdom of living creatures coping with changes in the environment: either adjust accordingly to adapt to the environment, or have some kind of robust nature that can accommodate any perceivable outcoming.

1.2.1 Knight Adaptive

An example of the unpredictable nature of channel can be easily found in wireless cell phone systems. Users of mobile phones are moving around, exiting one cell and entering another, their line of sight distance to the base station is changing all the time. The faithful and diligent power control system will be busy increasing or reducing the emission power of handset and base station to maintain a radio link of stable power level between them. This will generate a ever-changing inter-user interference effect on other users in the same or adjacent cells. When the mobile user roams the edge of a cell or the intersection of different cells, this kind of interference to other users can be so strong that it will cause annoying glitches and occasional drop-offs for mobile users.

Besides, there are also instabilities of equipment either in the network or at the user side, due to environmental heat and humidity, weather, component aging, malfunction, failing and software runaway. They will also introduce some uncertain elements to the communication system.

In these cases, we would like to implement some kind of adaptive scheme in the communication system to track the current parameters of the system model so that the receiver is always in tune and work in the optimal matched condition.

The adaptive receiver can combat lots of these uncertain elements.

1.2.2 Knight Robust

But sometimes even adaptation cannot solve the problem, therefore a robust receiver structure is needed for the worst possible condition.

In wireless radio channels, the environmental noise in the channel can normally be approximated as additive white Gaussian noise for practical purposes of designing communications systems. But some natural phenomenon such as lightening can cause impulsive noise. There is also a lot of man-made noise in the channel, such as automotive ignition noise, power-line noise and etc, which changes from place to place and time to time [50], [37]. The characteristic of one type of man-made noise is impulsive with a typical rate of 10-50 impulses/second, [50]. For a mobile phone with a data rate of 10 kbits/s, it could experience up to one impulse every 200 bits (roughly every speech packet will be affected). For a high frequency radio with a lower data rate of 1 kbits/s, the situation will be worse. Some types of man-made noise can be approximated by Gaussian noise, while others might only be modelled as other types of noise (for example, Laplace noise) [30]. Furthermore, even in Gaussian channels the maximum *a posteriori* probability decoder needs to estimate the noise variance. The accurate estimation of these noise parameters could range from being very difficult to impossible in practice.

In a word, the design of the two "White Knights" (adaptive and robust) for these powerful algorithms to overcome the adverse effect of uncertainty in the communications system is the primal focus of this thesis.

1.3 Overview of the thesis

1.3.1 Contributions

The major contributions of our research work are:

- Propose the new adaptive structure combining the joint adaptive scheme with the whitened matched filter and M-algorithm, which is both well performing and simple to implement.
- Apply the adaptive structures not only to maximum likelihood sequence detection in inter-symbol interference channel but also multiuser detection in code division multiple access systems.
- Provide useful guidelines for communication engineers in designing adaptive Viterbi algorithm or M-algorithm backed receivers; compared merits and disadvantage of the various adaptation combinations.
- Study the optimal robust detecting problem in a generalized way so that it can provide insight into extending the robust algorithms into various other areas, which will open a rich vein of related research.
- Apply the minimax robust kernel to the maximum likelihood sequence detector Viterbi algorithm, which is widely used in convolutional code decoding.
- Apply the minimax robust kernel to various *a posteriori* probability (APP) algorithms such as the decoders for Turbo code and low density parity check code, aiming to improving the robustness of decoders for these codes in uncertain channels or channels hard to be estimated accurately.
- Tackle the problem of mixed noise within one transmitted packet using our robust algorithms with success, which is not yet reported in any other robust decoding literature.

- Analyze the error performance of our minimax robust decoder and find the robust decoder has superior performance over the mismatched decoder and can perform very close to the optimal matched decoder. This conclusion is also supported by the numerical simulation results.

- Conclude from our analysis and experiment that our robust kernel is very easy to be implemented on top of the traditional decoder. And there are no or little computational complexity increase due to the introduction of the robust kernel. These are excellent characteristics for practical deployment.

1.3.2 Publications

- **"Breadth-First Algorithm with Adaptive Forney Structure for ISI channels"**, Zheng Li and Lei Wei, *Proceedings of Global Communication Conference (GlobalCom'98)*, Sydney Australia, Nov. 8-12 1998.

- **"On Robust Decoding Algorithms"**, Zheng Li, Lei Wei, Matthew James and Ian Petersen, Proceedings of *International Conference in Communications (ICC'99)*, Vancouver Canada, June 7-10 1999.

- **"A Minimax Robust Algorithm"**, Lei Wei, Zheng Li, Matthew James and Ian Petersen, accepted by *IEEE Transaction on Information Theory.*

1.3.3 Thesis architecture

The structural diagram of the thesis is shown in Fig.1.1.

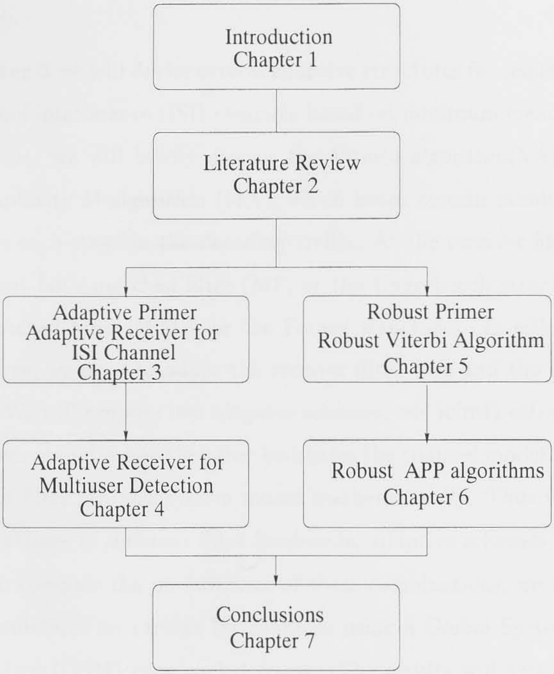


Figure 1.1: Structural diagram of the thesis

1.3.4 Chapter summary

Below is an outline of each chapter:

Chapter 2 briefly overview the various algorithms and system we will study in this thesis. It starts with Viterbi algorithm and reduced-complexity Viterbi algorithm, with application in convolutional code decoding, maximum likelihood sequence detection in inter-symbol interference channel and multiuser detection in CDMA systems. Then we discuss various *a posteriori* probability algorithms used in Turbo decoding and low density parity check code decoding. The state of art of both adaptive receiver design and robust decoding techniques are reviewed, in the context of the above powerful algorithms. This is the start point of our research work.

In **Chapter 3** we will devise several adaptive structures for sequence detection in inter-symbol interference (ISI) channels based on minimum mean square error criterion. First, we will briefly discuss the Viterbi algorithm(VA) and various reduced complexity M-algorithm (MA), which keeps certain number of survival paths (m) on each stage in the decoding trellis. At the receiver filter front-end, we will present both matched filter (MF, or the Ungerboeck structure) and the whitened matched filter (WMF, or the Forney structure). In either of the two filter structures, we need to adapt the receiver filter taps and the system model parameters. We will propose two adaptive schemes, one jointly estimate the filter tap and system model, while the other estimates the channel model first and then calculates the filter tap and system model mathematically. Thus we have a full set of combinations of different filter front-ends, adaptive schemes and sequence detectors. To compare the performance of these combinations, we will carry out extensive simulations on various ISI channels using a Global System for Mobile Communications (GSM) type packet frame. The results will provide us with a guide on how to choose from these adaptive structures for different purposes and some insight on why they will differ.

Chapter 4 will extend the adaptive structure to the multiuser detection problem for Code Division Multiple Access (CDMA) systems. This is a very natural extension because the multiuser detection problem and the ISI channel sequence detector problem share the same kind of trellis structure when using the Viterbi algorithms. The main difference will be different users instead of adjacent bits. We will focus on a synchronous CDMA system while the system is readily portable to asynchronous CDMA system. Again we will carry out simulations on the whole set of combinations of filter structures(MF or WMF), adaptive schemes(joint or separate) and sequence detectors(VA or MA). The results will help us choosing various adaptive structures to achieve a good performance and complexity compromise when designing multiuser receiver for CDMA systems

In **Chapter 5** we will propose the minimax concept and devise our robust receivers. The minimax idea, simply put, tries to optimize for the worst possible scenario. In our case, it means how to minimize the maximum error probability associated with all possible noise types. We approach the problem from a generic viewpoint, aiming to find the optimal robust algorithm. Although it provides us with a theoretical framework for designing robust algorithms, it appears to be too complicated even for a very simple problem. Our simplified near-optimal minimax robust algorithm is then proposed based on both the optimal robust algorithm and intuitive observation. After refinement, this robust kernel is injected into the Viterbi algorithm convolutional code decoder. There is generally no or slight computational overhead for adopting the robust scheme, and there are only a few structural adjustments needed in updating corresponding traditional receivers. All of the above attributes are very attractive for practical implementation of our robust algorithms. Our robust decoders generally outperform the mismatched decoders and is always very close to the optimal matched decoder. Our performance analysis also supports such observations.

Chapter 6 further extends the application of robust scheme into various *a posteriori* probability (APP) algorithms. We start with the Bahl, Cocke, Jelinek and Raviv 's algorithms [8] for Turbo code decoding, followed by the Gallager and McKay's decoding algorithms (GL/MN, [29],[44]) for low density parity check code. Our robust algorithm can handle mixed noise within a transmitted packet as well as across multiple packets, which is a unique contribution compared to other noise estimation methods appeared in the literature. Our simulation results will again illustrate the performance advantage we can achieve by improving the traditional APP algorithms with minimax concept. This performance edge, together with the implementation efficiency, will surely make our robust scheme an interesting new technique in improving lots of the powerful decoding algorithms widely used in today's error control coding systems.

Chapter 7 concludes the thesis by highlights of various contributions we made throughout the thesis. The purpose of our research in adaptive and robust receiver design is justified by our achievements. Finally the possible future works in this vibrant research area are presented.

Chapter 2

Literature Review

"...See human beings as though they were in an underground cave-like dwelling with its entrance, a long one, open to the light across the whole width of the cave. They are in it from childhood with their legs and necks in bonds so that they are fixed, seeing only in front of them, unable because of the bond to turn their heads all the way around. Their light is from a fire burning far above and behind them. Between the fire and the prisoners there is a road above, along which we see a wall. built like the partitions puppet-handlers set in front of the human beings and over which they show the puppets".

"Then most certainly," I said, "such men would hold that the truth is nothing other than the shadows of artificial things."

- The Cave from Book VII of Plato's Republic

It would be very helpful to first cruise through some important literature to better our understanding of the various topics we will cover in this thesis. The main thread is along powerful algorithms, adaptive receiver and robust decoding techniques.

2.1 Viterbi algorithm, M-algorithm and Applications

2.1.1 Viterbi algorithm

The Viterbi algorithm was originally proposed by Andrew J. Viterbi for decoding convolutional codes [79] [80] [81]. It is a recursive optimal solution to the problem of estimating the state sequence of a discrete-time finite-state Markov process observed in memoryless noise. It can achieve asymptotically optimal performance in additive white Gaussian noise channel. The idea is also known as "dynamic programming" in operations research.

We can map the discrete-time finite-state Markov process into a *trellis*, with each node standing for a distinctive state at a given time, each vertice or branch representing the legitimate transition from one state to another in the time sequence.

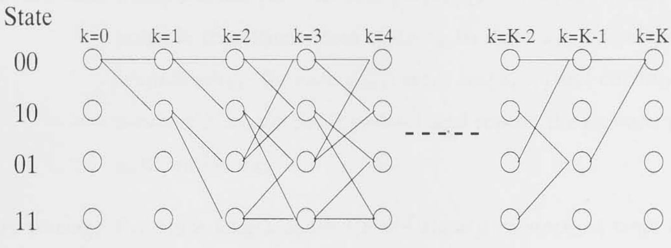


Figure 2.1: Trellis structure for Viterbi Algorithm

Suppose $\mathbf{x} = [x_1, x_2, \ldots, x_K]$ (K is the length of $\mathbf{x}$) is the vector of states in the trellis, $\mathbf{r}$ a sequence of observations contaminated by channel noise $\mathbf{n}$, we can associate each path in the trellis with a metric proportional to $-\ln P(\mathbf{x}, \mathbf{r})$. This way, we could solve the problem of finding the $\mathbf{x}$ to maximize $P(\mathbf{x}|\mathbf{r})$, by finding the path whose path metric $-\ln P(\mathbf{x}, \mathbf{r})$ is minimum.

$$\begin{aligned} P(\mathbf{x}, \mathbf{r}) &= P(\mathbf{x})P(\mathbf{r}|\mathbf{x}) \\ &= \prod_{k=0}^{K-1} P(x_{k+1}|x_k) \prod_{k=0}^{K-1} P(r_k|x_{k+1}, x_k). \end{aligned} \tag{2.1}$$

$$\begin{aligned} -\ln P(\mathbf{x}, \mathbf{r}) &= \sum_{k=0}^{K-1}(-\ln P(x_{k+1}|x_k) - \ln P(r_k|x_{k+1}, x_k)) \\ &= \sum_{k=0}^{K-1} bm_k \end{aligned} \tag{2.2}$$

in which bm_k stands for the branch metric at time slot k.

And this standard shortest-path problem can be solved by the following steps:

Initialization: Start with x_0 state, set the state metric $sm(x_0)$ to zero and all other state metric to infinity.

Forward recursion: Compute the state metric $sm(x_{k+1}) = sm(x_k) + bm_k$ for all possible transitions from state x_k to state x_{k+1}; find the minimum of sm_{k+1} for each x_{k+1}; store $sm(x_{k+1})$ and corresponding survivor $\hat{x}(x_{k+1})$. Set k to $k+1$ and repeat the forward process until we hit x_K.

Backward tracing: For finite length sequence the algorithm stops at time K with the shortest complete path, we can then retrieve this path from the stored survivor $\hat{x}(x_K)$

2.1.2 Reduced-complexity VA

Viterbi algorithm is attractive for its asymptotical optimal performance. One of its major drawback, however, is the large memory and computation, which is exponential in the duration of ISI or the constraint length of convolutional code.

It's the complexity issue of Viterbi algorithm that prevents it from ubiquitous deployment in practical communication system.

In light of this, several reduced-complexity near-optimal Viterbi algorithms are proposed, such as M-algorithm and T-algorithm [4] [87] [7] [5]. The major focus is on how to reduce the number of survival paths kept throughout the forward recursions.

For the M-algorithm, the best m extended paths are kept as survivors. For T-algorithms, only those paths whose path metric is within a threshold of T from the best path is stored. A combined M- and T- algorithm, or hybrid MT-algorithm, is a T-algorithm whose number of survivor paths are further restricted to m. If the $m = 2^{K-1}$ or $T = \inf$, then both algorithm will become the optimal Viterbi algorithm. On the other extreme, if $m = 1$ or $T = 0$, then these algorithms is no difference from decision feedback detector. From the above discussion, we can conclude that these reduced-complexity Viterbi algorithms are fine tunable to accommodate different tradeoffs in complexity and performance.

And there are strong analysis and simulation results suggesting that these significantly simplified algorithms can achieve near optimal performance [83] [84].

Instead of reducing the surviving paths in the full-sized trellis, other reduced complexity techniques try to reduce the number of states of the trellis, such as reduced state sequence detection [16] [20] [19], and using a linear or decision feedback equalizer to shorten the ISI duration [9] [21] [38] [54] [85] [31].

2.1.3 Application of VA and MA

The application of Viterbi algorithm in convolutional code is straightforward. Other researchers extend Viterbi algorithm to a wider variety of applications, such as sequence detection in inter-symbol interference channel [24] and multiuser detection for code division multiple access system [78] [75] [76].

For convolutional code and ISI channel, signals are correlated in the time space, or the transmitted signal s_k is a linear combination of information signals $[b_k, b_{k-1}, b_{k-2}, \ldots, b_{k-L}]$ (L is the memory length). For convolutional code, the correlation will be either 0 or 1; for ISI channel, the correlation factor will be the channel taps. The information sequence $[b_k, b_{k-1}, b_{k-2}, \ldots, b_{k-L}]$ corresponds to the state or node in the trellis graph, and the branch metric could be computed in proportion to $-\ln P(r_k|b_k)$ where $r_k = s_k + n_k$.

For synchronous CDMA systems, the correlation exist among different simultaneously transmitting users in the code space. Another level of time space correlation is added for asynchronous CDMA systems.

2.2 *A posteriori* probability algorithms

Recently, the two-way *a posteriori* probability (**APP**) decoder has attracted lots of attention, due to its application in Turbo decoding [8] [10] and Low-Density Parity Check (**LDPC**) decoding [28] [29] [42]. Both the Viterbi algorithm and two-way APP algorithms are special cases of min-sum and sum-product algorithms [86]. Some significant developments in understanding these two-way decoding algorithm have been reported in [69], [43], [26], [6]. In [25] Forney provided a detailed discussion of the two-way algorithms as well as an overview of their rich history.

The TWL (Tanner, Wiberg, Loeliger) graph provides us with a convenient way of visualizing various coding schemes and insights into the comparative merits and shortcomings of those code structure and their decoding algorithms. In TWL graphs, nodes represent symbols, state and checks. The relation between nodes are legal state transition or parity check constraints. The two-way min-sum and sum-product algorithms are straightforward and optimal solutions to certain decoding problems for finite cycle-free TWL graph.

The practical decoding algorithms for most powerful codes, such as Turbo code, LDPC code and tail-biting code can be derived from min-sum and sum-product algorithms taking into account that these codes are inherently represented by graph with cycles. They work very well as long as the cycles are long enough that cyclical dependencies die out as they propagate around a cycle. [25]

Besides these optimal two-way algorithms, there are several suboptimal forward only algorithms developed for reducing the complexity [3] [2] [89].

2.3 Adaptive detection

2.3.1 Adaptive equalization for ISI

There are lots of research work on adaptive equalizer design for ISI channel [27] [53] [55], and we could categorize them into adaptive linear equalizer, adaptive decision-feedback equalizer and adaptive channel estimator for maximum likelihood sequence detection (Viterbi algorithm). We will skip over the first two and focus our attention on the last category due to its comparatively good performance.

Ungerboeck [70] proposed an adaptive matched filter front-end coupled with Viterbi algorithm sequence detector structure, which could simultaneously adjust the demodulating carrier phase and sample timing, approximate the matched filter by a transversal filter, and estimate ISI present at the output of the approximated matched filter. Stochastic steepest-descent algorithms is used to derive the recursive adaptation steps. It differs from Forney's whitened matched filter structure [24] with its matched filter front-end and adaptive structure.

As we discussed in the previous section, maximum likelihood sequence Detector(MLSD) implemented by Viterbi algorithm has exponential complexity in regard to the memory length. Therefore, some receiver utilize an adaptive equalizer to constrain the length of the equivalent channel impulse response. In [31], Gu proposed an embedded decision feedback equalizer(DFE) to act simultaneously as a pre-whitening matched filter, a compensator for channel distortion and an adaptive equivalent channel impulse response estimator while the embedded MLSD detector operates on the signals predicted by the embedded DFE. This could adaptively trace the simplified channel model.

2.3.2 Adaptive multiuser detection for CDMA

The multiuser detector for CDMA systems depends on various system parameters such as received signal amplitude and cross-correlations which are fluctuating both in time and space. Therefore, the self-tuning adaptive multiuser detection attracted much interest in recent years.

Verdu has a good overview on adaptive multiuser detection in [77]. In brief, there are decorrelating detector [39] [40], linear multiuser MMSE detector [88] [41] [56], tentative-decision based detector [72] [73] [15], blind multiuser detector and Neural network based detector.

There are certain degree of similarity between the adaptive multiuser detector and adaptive ISI receiver, provided the detector end use Viterbi algorithm. This is because each user at each correlating chip in the asynchronous CDMA system and each interfering symbol period of ISI have similar representation in the Viterbi algorithm. However, the multiuser detector is normally more complicated because of the longer correlation length in heavily loaded system. The fundamental adaptive structure and system model, though, remain comparable.

2.4 Robust decoding

2.4.1 Non-Gaussian noise

Although the assumption of white Gaussian noise is quite appropriate for many applications, it is well known that in many practical channels the noise distribution can be hardly modelled as Gaussian due to the existence of various impulsive noise [82] [1] [37]. This is particularly true in urban and indoor radio channels for mobile and portable communications. For detailed report on the measurement and modelling methods please refer to [12] [13] [45] [46] [51] and the references therein.

Non-Gaussian impulsive noise can be quite detrimental to the performance of traditional detector based on Gaussian assumption. On the other hand, a properly modelled and estimated noise model can be quite beneficial to the detector design. There have been numerous efforts over the past three decades in the area of signal detection in impulsive noise [34] [47] [48] [65] [66] [67] [74].

2.4.2 Robust decoding based on noise estimation

Several recent works use robust estimation method to address the problem of Turbo decoding in channels with unknown noise PDF. Summers and Wilson in [68] and Reed and Asenstorfer in [58] focus on how to efficiently and accurately estimate the noise variance of each block for Turbo decoders. Huang and Phamdo in [32] deals with how to accurately estimate the noise distribution within a family of noise models. This is still a relatively new research area which is generating more and more interest from both academia and industry.

Both Summers and Reed found that there is performance degradation for MAP algorithm Turbo decoder if the channel noise variance is not correctly estimated, as shown in Fig.1 of [68] and Fig.2 of [58]. Another interesting finding is that over-estimating noise variance is less detrimental than under-estimating variance, tolerating a mismatch of several decibels without significant degradation. The reason behind this phenomenon is not presented, though.

Summers devised a blind algorithm to estimate the unknown SNR from each code block prior to decoding that block, which do not require the transmission of training symbols. He used a heuristic approach which is based on sums of the squared receiver values and sums of their absolute value. This online estimation method do not degrade performance appreciably relative to the known SNR conditions.

Reed proposed a Novel Variance Estimation Technique (NOVEL) based on the assumption that the output of Turbo decoder is approximately equal to the data sent. This variance estimation method is however one block behind that of the conventional techniques because the decoder result is required before the estimation can be made. The performance analysis and simulation results show NOVEL is better than conventional algorithms at low SNR region, where the Turbo code is normally used.

Huang [32] investigated the sensitivity of Turbo decoder to the noise distribution mismatch and proposed a simple on-line estimator for each block of received signal. The estimator first quantize the received signal and then estimate the noise distribution from the histogram of the quantized received signal. They achieved decoder performance within 0.1dB at BER 10^{-4} from the exactly known case.

Chapter 3

Adaptive receiver for ISI channel

Nothing is permanent but change.
- Heraclitus

3.1 The purpose

The inter-symbol interference (ISI) can seriously affect the performance of receiver in wireless and wireline communication systems so it's important to devise adaptive schemes to combat this problem.

There are lots of research work on adaptive receiver design for ISI channel. However little previous work are focused on the reduced-complexity Viterbi algorithms or more specifically the M-algorithm. And we know that M-algorithm can achieve near optimal performance at significantly low complexity [83] [84].

In this chapter, we will start with a primer on ISI channel, followed by details of matched and mismatched filter front-ends, Viterbi algorithm and M-algorithm detector back-ends, jointly adaptation scheme 1 and channel estimation scheme 2, then arrive at a full combination of adaptive structures. The simulation results will give us insight into how to choose among these combinations when designing real systems.

3.2 Sequence detection in ISI channel

3.2.1 What is inter-symbol interference (ISI) channel?

So what is inter-symbol interference channel, also know as ISI channel? Let's first consider a simple analogy:

Suppose you are in the mountains, you start to recite Homer's Iliad to show your friends your talents as an orator. Your voice is loud and clear and you are about to make an impression, however, something happens. The reverberant sound of yourself several seconds ago bounce back from the mountains and add right on top of your current recital, making your friends hard to distinguish any word, let alone admire your talent. So what's the problem? Sound travels in different paths which result in different delays in reaching the listener, and they get mixed up with the original sound.

Inter-symbol interference channel has the similar mechanism. Due to the different paths source signal traverse, various signals arrive at the receiver with different time delay (or phase shift) and amplitude attenuation. This phenomenon frustrates telecommunication engineer because the original signal is not only contaminated by the environmental noise, but also by several distorted copies of itself, which further complicates the detection task.

3.2.2 System model

Before we move onto the solution to the inter-symbol interference problem, let's first formulate it. The system block diagram of the transmitter and receiver over typical ISI channel is shown in Fig.3.1.

The signal at the output of the channel is

$$r(t) = \mathbf{h}(t) * \mathbf{b}(t) + n_0(t), \tag{3.1}$$

where $*$ denotes convolution, $r(t)$ is the received signal, $h(t) = p(t) * C(t)$ is

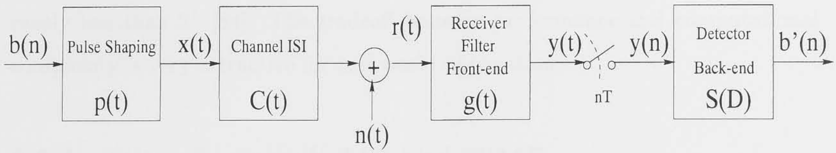


Figure 3.1: System block diagram of ISI channel

equivalent channel impulse response, $p(t)$ is the pulse shaping waveform, $C(t)$ is the channel impulse response, L is the number of channel ISI taps, $\mathbf{b}(t)$ is information source bits and $n_0(t)$ is additive white Gaussian noise with zero mean and variance σ^2.

Transition from continuous to discrete time model using the delay operator D (D-transform) is discussed by Forney in [24], in order to form a set of sufficient statistics for estimation of input sequence $\mathbf{b}$.

3.2.3 Maximum likelihood sequence detection

A good way to combat the inter-symbol interference is to intelligently use the additional information contained in the interference from adjacent bits to cancel out its effect . Therefore we will detect each symbol in the context of an interfering sequence instead of the single received signal that symbol represents. Otherwise the inter-symbol interference will be regarded as part of the environmental noise, which in turn result in a lower signal to noise ratio and poorer performance because noise is enhanced.

The optimal maximum-likelihood sequence detector try to select the most likely data sequence out of all the possible sequences using Viterbi algorithm. However the complexity of VA grows exponentially with the memory size L for inter-symbol interference channels, which could be prohibitive for large L. The M-algorithm limits the surviving states at any time interval to m[4]. It was found that the M-algorithm can approach near optimum performance using m significantly less than 2^L [84]. The tradeoff between performance and computational complexity is very attractive for real world application.

3.2.4 Filter front-end: MF and WMF

There are two basic types of receiver filter for ISI channel: one is the Ungerboeck's matched filter structure; another one is the Forney's whitened matched filter structure.

Matched filter (Ungerboeck structure)

The matched filter (in this chapter we restrict to real signals) can be expressed as

$$\mathbf{g}_{MF}(D) = \mathbf{h}(D^{-1}). \tag{3.2}$$

The signal after the matched filter will be

$$\begin{aligned} y(D) &= \mathbf{h}(D^{-1})\mathbf{h}(D)\mathbf{b}(D) + \mathbf{h}(D^{-1})\mathbf{n}(D) \\ &= \mathbf{R}(D)\mathbf{b}(D) + z(D), \end{aligned} \tag{3.3}$$

where $\mathbf{n}(D)$ is white Gaussian noise, $\mathbf{R}(D) = \mathbf{h}(D^{-1})\mathbf{h}(D)$ is the system correlation matrix and $z(D)$ is a coloured Gaussian noise sequence, whose correlation matrix is $\mathbf{R}(D)$.

Using the maximum likelihood criteria, as shown in [70], we can get the metrics for the Viterbi algorithm

$$\begin{aligned} \mathbf{J}_n(b'_0, b'_1, \ldots, b'_{n-1}, b'_n) = \mathbf{J}_{n-1}(b'_0, b'_1, \ldots, b'_{n-1}) \\ + b'_n(2y_n - S_0 b'_n - 2\sum_{l=1}^{L} S_l b'_{n-l}). \end{aligned} \tag{3.4}$$

Here $\mathbf{J}_n$ is the state metrics ending symbol bit b'_n; the second part of the right hand side of the (3.4) is the branch metric; y_n is the filtered signal to be fed into the sequence detector; S_l is the system pulse response parameters for the matched filter structure. Because of the symmetrical characteristics of the matched filter structure, $S_{-l} = S_l$, only one side is needed to calculate the metric.

Whitened matched filter (Forney structure)

In [24], it was shown that after the whitening filter $\mathbf{g}_{WF} = (\mathbf{F}(D^{-1}))^{-1}$, where $\mathbf{R}(D) = \mathbf{F}(D)\mathbf{F}(D^{-1})$. We have

$$\begin{aligned} y'(D) &= \mathbf{g}_{WF}(D)y(D) \\ &= \mathbf{F}(D)\mathbf{b}(D) + n'(D) \\ &= \mathbf{g}_{WMF}(D)\mathbf{b}(D) + n'(D), \end{aligned} \tag{3.5}$$

$$\begin{aligned} \mathbf{g}_{WMF}(D) &= \mathbf{g}_{WF}\mathbf{g}_{MF} \\ &= \mathbf{F}(D^{-1})^{-1}\mathbf{h}(D^{-1}), \end{aligned} \tag{3.6}$$

where $n'(D)$ is whitened Gaussian noise sequence.

According to [24], the metric for the maximum likelihood criteria could be computed as following

$$\mathbf{J}_n(b'_0, b'_1, \ldots, b'_{n-1}, b'_n) = \mathbf{J}_(b'_0, b'_1, \ldots, b'_{n-1}) + (y_n - \sum_{l=0}^{L} F_l b'_{n-l})^2, \tag{3.7}$$

where F_l is the system impulse response of the whitened matched filter structure.

3.3 Adaptive Structure

3.3.1 Adaptation Schemes

Here we will consider two groups of adaptation schemes:

Adapt 1: jointly adapt receiver filter matrix g and system correlation matrix S;

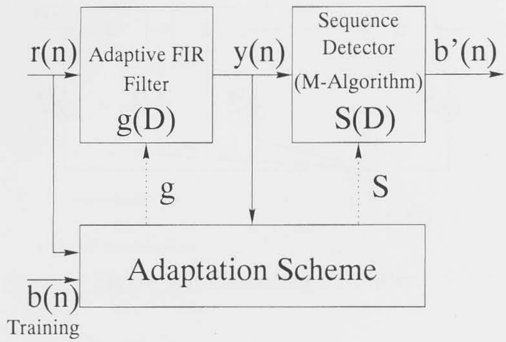


Figure 3.2: Adaptation structure for ISI channel

Adapt 2: adaptively estimate channel impulse response h and then compute g and S from h.

Adaptation structure is illustrated in Fig. 3.2.

Adaptation Scheme 1

Let's first consider the jointly adaptive filter structure. In [70], only adaptive matched filter was given. In this thesis, we present both adaptive matched filter and adaptive whitening matched filter. The finite impulse response filter is shown in Fig. 3.3.

The system can be expressed as

$$y_n = \mathbf{g}^T \mathbf{r}_n, \tag{3.8}$$

where $\mathbf{g} = [g_0, g_1, \ldots, g_K]^T$ is the receiver FIR filter taps and K is the number of the FIR filter taps. For matched filter, $K = L$; for whitened matched filter, $K \geq L$ depended on receiver cut-off from infinite impulse response filter.

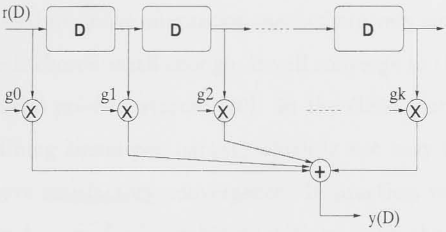


Figure 3.3: Cascading FIR filter

The receiver tap coefficients $\mathbf{g}$ and system impulse response $\mathbf{S}$ are estimated as $\hat{\mathbf{g}}$ and $\hat{\mathbf{S}}$ according to

$$\{\hat{\mathbf{g}}, \hat{\mathbf{S}}\} = \arg\min_{\tilde{\mathbf{g}}, \tilde{\mathbf{S}}} E\{\| \tilde{\mathbf{g}}^T \mathbf{r} - \tilde{\mathbf{S}}\mathbf{b} \|^2\}, \tag{3.9}$$

A simple recursive way to adjust the tap coefficients is the stochastic gradient method by differentiating 3.9 and applying the Robbins-Monro stochastic approximation method [60]. In [70], Ungerboeck presented the recursive scheme for matched filter front end, here we extend it further to whitened matched filter front end [24].

$$\begin{aligned} \hat{\mathbf{g}}^{n+1} &= \hat{\mathbf{g}}^{n} - \mu_g e_n \mathbf{r}_n \\ \hat{\mathbf{S}}^{n+1} &= \hat{\mathbf{S}}^{n} + \mu_S e_n \mathbf{b}_n \end{aligned} \tag{3.10}$$

$$\begin{aligned} e_n &= y_n - \hat{S}_0 b_n - 2\textstyle\sum_{i=1}^{L} \hat{S}_i b_{n-i} \quad \text{Matched filter} \\ e_n &= y_n - \textstyle\sum_{i=0}^{L} \hat{S}_i b_{n-i} \quad \text{Whitened matched filter} \end{aligned} \tag{3.11}$$

where e_n is the estimation error, $\hat{\mathbf{g}} = [\hat{g}_0, \hat{g}_1, \ldots, \hat{g}_K]$ is the estimated receiver FIR filter taps; $\hat{\mathbf{S}} = [\hat{S}_1, \ldots, \hat{S}_L]$ is the estimated system impulse response; $\mathbf{r}_n = [r_n, r_{n+1}, \ldots, r_{n+K}]$ is the received signal; $\mathbf{b}_n = [b_n, b_{n-1}, \ldots, b_{n-L}]$ is the training source sequence; μ_g is positive step size for adapting $\hat{g}$; μ_S is positive step size for adapting $\hat{S}$.

The gradient based recursive adaptation method are very simple to implement. When the step size is chosen small enough, it will converge to the global optimum no matter what initial point it started [70]. In the GSM system [57], there are only 26 bits of training signal per packet, which is not easy for gradient based algorithms to achieve satisfactory convergence. In practice, we find by choosing step size $\mu_s \approx \frac{1}{L}$ and $\mu_g \approx \frac{1}{10}\mu_s$, combining with re-using the training bits, this joint adaptation scheme can yield good results in typical noise environment.

Adaptation Scheme 2

An alternative adaptive scheme will be estimating the equivalent channel model $\mathbf{h}$ in (3.1) first and then calculate $\mathbf{S}$ and $\mathbf{g}$ from $\mathbf{h}$, which we denoted as adaptation scheme 2. Similar schemes has been used to compute other equalizer (LE/DFE) coefficients such as [22].

The channel impulse response $\mathbf{h}$ can be estimated as $\hat{\mathbf{h}}$ according to

$$\hat{\mathbf{h}} = \arg\min_{\tilde{\mathbf{h}}} E\{\| r - \tilde{\mathbf{h}}\mathbf{b} \|^2\} \tag{3.12}$$

where $[\hat{h}_1, \ldots, \hat{h}_L]$ is the estimated channel impulse response.

The recursive adaptation of $\hat{h}$ can be obtained using stochastic gradient algorithm

$$\begin{aligned} \hat{\mathbf{h}}^{n+1} &= \hat{\mathbf{h}}^n + \mu_h e_n \mathbf{b}_n, \\ e_n &= r_n - \textstyle\sum_{i=0}^{L} \hat{h}_i b_{n-i} \end{aligned} \tag{3.13}$$

where e_n is the estimation error, $\hat{\mathbf{h}} = [\hat{h}_0, \hat{h}_1, \ldots, \hat{h}_L]$ is the estimated channel impulse response and μ_h is positive step size for adapting $\hat{h}$.

After we get the estimate of channel coefficients $\hat{\mathbf{h}}$, the system correlation matrix could be calculated by:

$$\hat{\mathbf{R}} = \hat{\mathbf{h}} * \hat{\mathbf{h}} \tag{3.14}$$

If we want to use the matched filter structure, we could simply use $\hat{\mathbf{g}}_{MF} = \hat{\mathbf{h}}$ and $\hat{\mathbf{S}}_{MF} = \hat{\mathbf{R}}$ as FIR filter and system impulse response; if we want to use the whitened matched filter structure, then we could use the window Cholesky decomposition ($\mathbf{R}(D) = \mathbf{F}(D)\mathbf{F}(D^{-1})$) to get the whitened matched filter system matrix $\hat{\mathbf{S}}_{WMF} = \hat{\mathbf{F}}$, which in turn could be used to get the receiver FIR filter

$$\hat{\mathbf{g}}_{WMF} = \hat{\mathbf{F}}^{-1}\hat{\mathbf{h}}. \tag{3.15}$$

This way, the noise whitening requirement can be guaranteed by the Cholesky factorization process. But this could also be costly due to the additional matrix computation.

3.3.2 FIR filter front-end and adaptation scheme combinations

According to the above discussion on "receiver FIR filter structure"and "Adaptation schemes", it is quite clear that we have a few interesting combinations to study, which is shown in the following graph:

$$\begin{pmatrix} \text{MF} \\ \text{WMF} \end{pmatrix} \begin{pmatrix} \text{Fixed} \\ \text{Adapt1} \\ \text{Adapt2} \end{pmatrix} \begin{pmatrix} \text{VA} \\ \text{MA(m)} \end{pmatrix}$$

The abbreviations used in the graph are listed as follows:

MF: Matched filter;

WMF: Whitened matched filter;

VA: Viterbi algorithm;

MA(m): M-algorithm (m surviving states).

The adaptive whitened matched filter schemes coupled with M-algorithm is of great interest to us because they are the hope for achieving near optimal performance at a very low receiver complexity.

3.4 Simulations results

In this section, we study the different adaptive receiver structures under a GSM style environment. The data packet structure is similar to that in the GSM system (140 bits per packet, of which 26 are training bits); pulse shaping is rectangular pulse. The channel has 3 taps $[0.407, 0.815, 0.407]$ [52], memory length $L = 2$, i.e. the total number of states is 4. In the graphs, F denotes channel parameters known and fixed; A1 and A2 denotes adaptive scheme 1 and 2 respectively. The simulation results are in Figs 3.4(a) and 3.4(b).

We also did simulation on a 5 tap ISI channel $[0.2917, 0.4941, 0.5842, 0.4941, 0.2917]$, $L = 4$, in Figs 3.5(a) and 3.5(b).

3.5 Conclusions

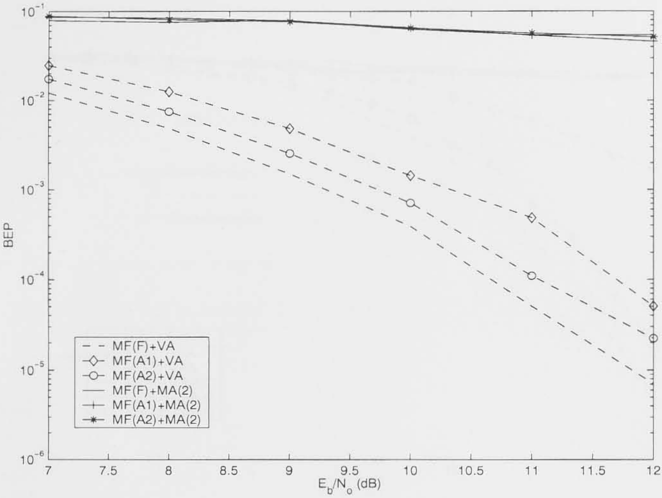
From the graphs we can find that if the channel parameters are known and fixed, with Viterbi algorithm, both the MF and WMF can get identical bit error performance. But when we use M-algorithm with a small m, the performance of matched filter structure degrade sharply compared with that of Whitened matched filter. For the latter, there is very minor performance degradation. So for low-complexity detector like M-algorithm, it shows that whitened matched filter generally performs better.

Comparing the two adaptation schemes A1 and A2 in the graphs, we will find that A2 generally yield better error performance than A1. It is not surprising because the A1 try to adapt $\mathbf{S}$ and $\mathbf{g}$ simultaneously with limited training length.

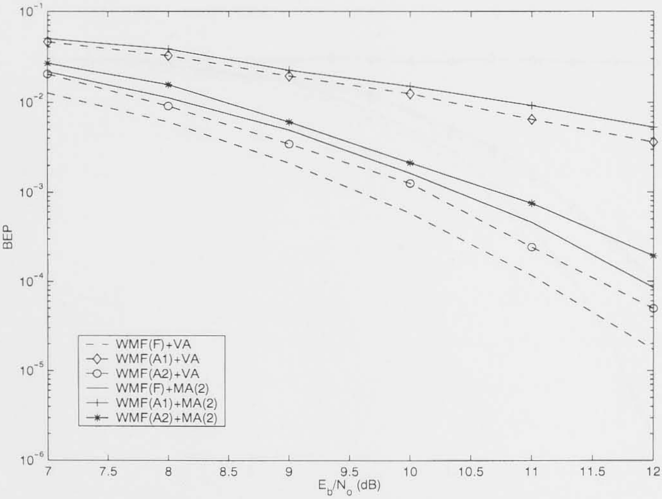
However the computational complexity of A2 (there are overhead for Cholesky decomposition and matrix multiplication) is much greater than that of A1. This overhead is averaged among the information carrying bits (or pay load) in a packet, so it would be quite significant for small packets or moderate for longer packets.

Another point worth noting is that the whitened matched filter using joint adaptation scheme A1 is somewhat disappointing, mainly because we truncated the infinite impulse response filter to simulate the whitened matched filter. When the parameters are known and fixed, such truncation will have minimum impact on the system performance. However, in an adaptive setup, sensitivity related issue comes into play which has a negative impact on the orthogonality constraint. Improvement on this specific combination will be part of the future work.

In a word, there is no clear cut winner among these adaptive receiver structures. The above results aim to present a comparison matrix in helping design adaptive receiver based on performance and complexity requirements.

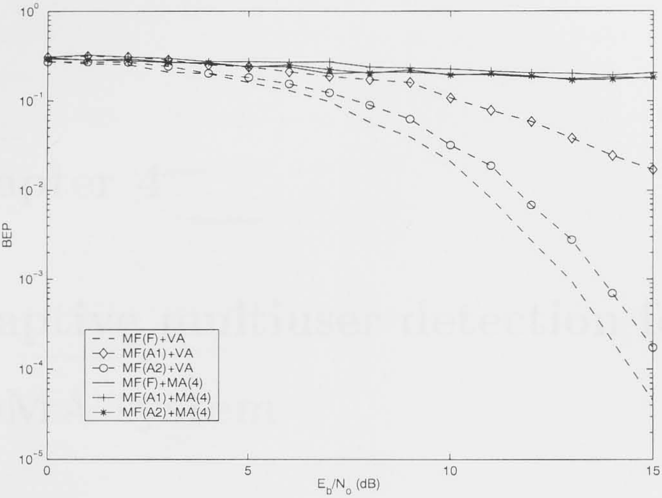


(a) Matched filter front-end

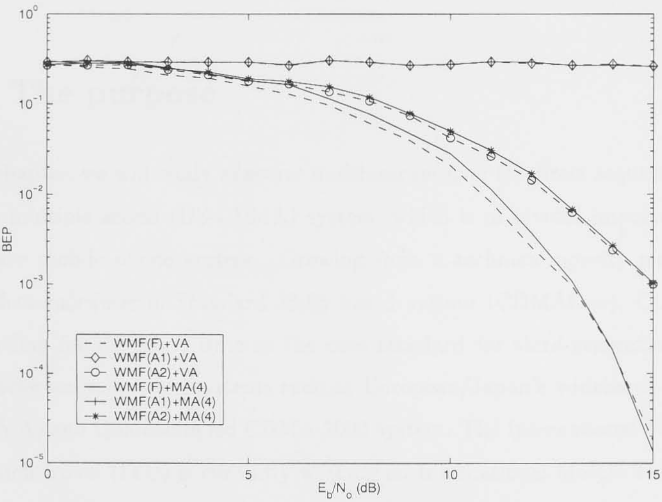


(b) Whitened matched filter front-end

Figure 3.4: Adaptive Sequential detection using different filter front-end; Channel $C = [0.407, 0.815, 0.407]$



(a) Matched filter front-end



(b) Whitened matched filter front-end

Figure 3.5: Adaptive Sequential detection using different filter front-end; Channel $C = [0.2917, 0.4941, 0.5842, 0.4941, 0.2917]$

Chapter 4

Adaptive multiuser detection for CDMA system

The noise is so great, one cannot hear God thunder.

- R.C. Trench

4.1 The purpose

In this chapter, we will study adaptive multiuser receiver for direct sequence code division multiple access (DS-CDMA) system, which is of pivotal importance to our future mobile phone system. Growing from a technical novelty pioneered by Qualcomm's Interim Standard IS-95 based system (CDMAOne), CDMA is well heading for its prime time as the core standard for third-generation (3G) mobile telecommunication systems such as European/Japan's wideband CDMA (W-CDMA) and Qualcomm led CDMA-2000 system. The International Telecommunication Union (ITU) is currently working on International Mobile Telephone system for 2000 (IMT-2000) to accommodate various proposals in hope of reaching a universal global standard.

Our work is focused on designing receivers which has good compromise between performance and complexity and can adaptively adjust itself to reflect the variations in the communication system. The compromise is achieved by introducing a reduced complexity Viterbi algorithm- the m-algorithm, coupled with adaptive filter front-end. Various adaptation techniques are studied to estimate the channel and system parameters using a pilot training signal. Their comparative merits are evaluated side by side through numerical simulation.

4.2 Code Division Multiple Access system

4.2.1 Multiple access overview

Basically, there are three different kinds of multiple access or channelization technologies: frequency division multiple access (FDMA), time division multiple access (TDMA) and code division multiple access (CDMA). They separate different users of the trunk communication media from crosstalk by a different frequency band, time slot or spreading code Fig.4.1. Following is a brief overview of all these technologies within the context of mobile communication systems.

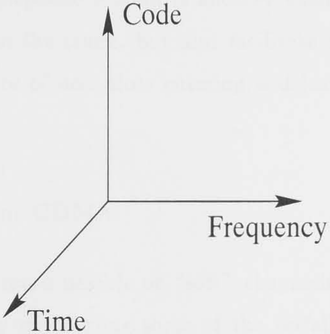


Figure 4.1: Multiple access philosophy

"Hard" Channelization: FDMA and TDMA

Both FDMA and TDMA are essentially "hard" resource sharing approaches. They have deep mentality roots in the permanent circuit oriented telecommunication systems, which dated as far back as Edison's telephone system. They allocate a slice of channel resource to each active user, who have the absolute and sole use of that resource during its session, with a predictable quality assurance.

Due to the precious nature of our frequency bandwidth, virtually all practical communication system use certain version of FDMA technology. Giving an example in the mobile phone arena: the first generation analog Advanced Mobile Phone System (AMPS), assigns a bandwidth of 30KHz for each mobile user. The Global System for Mobile Communication (GSM) slices the radio channel into 200KHz slots which is further shared by another layer of multiplexing: TDMA. [57]

The time division multiple access use the time domain diversity to segregate different users. The above mentioned GSM 200KHz frequency band actually is shared by 8 users, each occupying one of the eight time slots in a 4.615ms frame. (The digital switched telephone system is another example, in which the time slots not only channelize the trunk, but also facilitate the digital switching by re-arranging the sequence of user slots entering and leaving a switching fabric. [33])

"Soft" Channelization: CDMA

The CDMA scheme, a more flexible or "soft" channelization method, attracts increasing popularity by overcoming some of the pitfalls of traditional "hard" resource sharing methods. Here is an un-exhaustive list of CDMA's attractiveness in the context of wireless mobile phone systems:

1. **It scales better**: We can add on more new users to a fully-utilized CDMA system without significantly compromising service quality for existing users. This is a so-called "graceful performance degradation" or "Something for everybody" scenario. While in traditional "hard" multiple access systems, when all the channels are allocated, no further user can communicate, or an "Everything or nothing" scenario.

2. **It is more efficient**: CDMA system have native support for variable data rate and multiple Quality of Service (QoS). In a human to human conversation, more than 65 percent talk time is idle [37], so "hard" channelization will waste a lot of communication resources during non-talking period. For bursty packet data transmission, we need not only variable data rate but also different priority for different services.

3. **It can provide better quality**: Due to its spread spectrum technology, CDMA systems are inherently less prone to narrow-band interference and frequency-selective fading, which results in clearer voice and less drop-offs. In addition, soft hand-off between base-stations means you have less glitch or dropout for high mobility users when crossing cell boarders.

4. **It is easier to deploy**: No longer will we need complicated frequency planning. Now we can easily re-use frequency within a cell or among adjacent cells and implement hierarchical cell structure (microcells or picocells) to support "hot spot" (a place over-crowded by mobile terminals, e.g., airport and sports stadium) and semi-fixed broadband data transmission.

These merits are to die for considering the hard challenges facing telecommunication engineers in the wireless battlefield. That can partly explain why all the major vendors and standard bodies for wireless mobile phone systems, quarrelsome as they always are, more or less unanimously agree on using CDMA technology for the third generation international mobile phone standard. Next we will study the details of direct sequence CDMA technology.

4.2.2 Synchronous CDMA system

CDMA basics

The concept behind code division multiple access is profound. It might not be as instinctive as frequency or time segmentation, therefore we will start with a very simple example to show why using a spreading code to distinguish a user can work like magic.

Suppose we have two users sending information sequence b_1 and b_2, each of them is assigned a unique and mutually orthogonal code h_1 and h_2. (Fig.4.2)

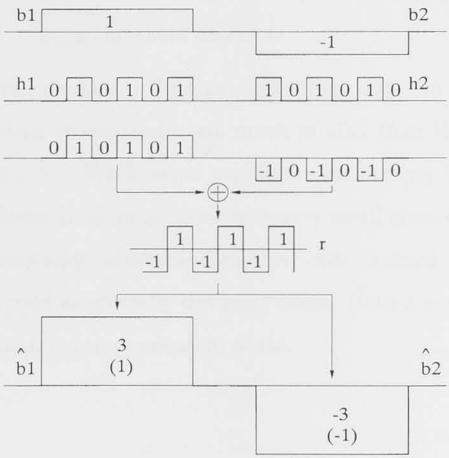


Figure 4.2: Code division multiple access basics

$h_1 = [0, 1, 0, 1, 0, 1,]$, $\quad h_2 = [1, 0, 1, 0, 1, 0]$

Each source symbol is spreaded into 6 chips using their code and then modula-2 added together for transmission. Suppose at t symbol slot user 1 transmit 1 and user 2 transmit -1, then the transmitted signal

$\mathbf{r} = h_1 b_1 + h_2 b_2 = [-1, 1, -1, 1, -1, 1]$.

At the receiver side, we use the matching code sequence to de-multiplex. For

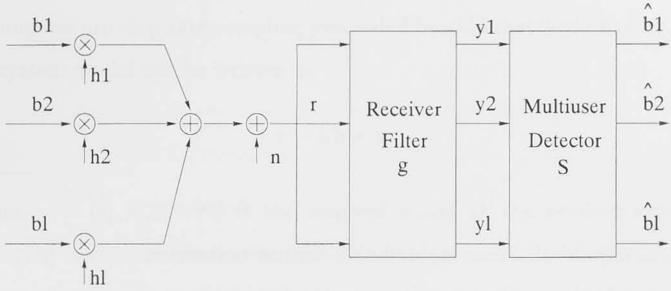


Figure 4.3: Synchronous CDMA system diagram

user 1, $\hat{b}_1 = r * h_1 = 3$ while user 2, $\hat{b}_2 = r * h_2 = -3$. It's quite clear that we could easily decode using a threshold detector.

In the above case, we used orthogonal spreading codes. In the real world, as long as the correlation among codes are much smaller than the autocorrelation of themselves, it is okay. While some people are still trying hard to search for large groups of orthogonal codes or codes with very small cross-correlation, others resort to generate very long pseudo random (PN) codes instead, arguing that their random code is as good as specially designed codes. Both approach have success stories in the real mobile communication world.

System Model

First we draw the system diagram of the baseband synchronous CDMA system, then proceed to define the mathematical model, which will be the foundation for the rest of this chapter.

In Fig.4.3, there are l users communicating simultaneously, and the spreading factor of the signature waveform is k. These two parameters are important because they decide the performance, capacity and complexity of CDMA systems. Here the spreading factor means how many chips the spreading code have for one symbol interval, e.g., $k = 6$ in the above example.

Assume we use chip-rate sampling preceeded by chip matched filtering, a simplified system model can be written as

$$\mathbf{r} = \mathbf{h}\mathbf{b} + \mathbf{n} \tag{4.1}$$

where $\mathbf{r} = [r_1, r_2, \ldots, r_k]$ is the received signal at the receiver side, $\mathbf{b} = [b_1, b_2, \ldots, b_l]$ is the information source of individual users; $\mathbf{h}$ (dimension $k \times l$) is the spreading matrix in which each column is the spreading code of corresponding user; $\mathbf{n} = [n_1, n_2, \ldots, n_l]$ is the channel noise at each chip slot.

We can clearly see now that the received signal $\mathbf{r}$ is a mixture of transmitted signals of all users and the channel noise. How to separate the desired user signal from other users and noise, is of no trivial task.

Single user and multiuser detection

If you treat the interference from other users as environmental noise, you end up with a single user detector; otherwise you would design some form of multiuser detector to fully utilize the information extracted from the inter-user interference.

The single user detector uses a band of filters each matched to the signature chip waveform of one user. After the filter bank, each signal is individually fed into a decision device to get the detection done. It's extremely simple to implement but there is significant performance loss due to the multiuser interference. In most practical wireless communication systems, these inter-user interference is much stronger than the environmental noise which makes the single user detector performs rather inefficiently.

There are many different types of multiuser detector, providing a rich combination of compromise between performance and complexity. There are de-correlator, minimum mean square error, multi-stage, successive cancellation, neural networks detector and etc. However, the best performing multiuser detector is Viterbi algorithm based which was originally proposed by Verdu in [78].

There are many less complicated variants of Verdu's optimal multiuser detector. Similar to our previous study, we found that m-algorithm, a breadth-first reduced-complexity Viterbi algorithm, is particularly attractive [83] [84]. So in the next section we will have detailed discussion of such kind of multiuser detectors.

4.2.3 Multiuser detection with Viterbi algorithm and m-algorithms

Refer back to Fig. 4.3, the signal after the receiver filter bank will be:

$$\mathbf{y} = \mathbf{gr} = \mathbf{Sb} + \mathbf{z} \tag{4.2}$$

where $\mathbf{g}$ (dimension $l \times k$) is the receiver filter matrix; $\mathbf{S} = \mathbf{hg}$ is the system correlation matrix; $\mathbf{z} = [z_1, z_2, \ldots, z_l]$ is the noise component after the filter bank.

The purpose of the detector, is to find the most likely transmitted signal set $\hat{\mathbf{b}}$ according to:

$$\hat{\mathbf{b}} = \arg\max_{\tilde{\mathbf{b}}} \mathrm{P}(\mathbf{y}|\tilde{\mathbf{b}}) = \arg\max_{\tilde{\mathbf{b}}} \prod_{i=1}^{l} \mathrm{P}(\mathbf{y}|\tilde{b}_i) \tag{4.3}$$

We can use the Viterbi algorithm to solve this problem by constructing the trellis in such a way that the node (or state) stands for the $\hat{\mathbf{b}}$ hypothesis, the vertices (or branch) stands for whether a certain user transmit 0 or 1. Associate each branch with the probability $\mathrm{P}(\hat{b}_i|\mathbf{y})$, we can work through the trellis to find the most likely $\hat{\mathbf{b}}$ for Eqn.4.3 [78].

The M-algorithm, one type of breadth-first reduced-complexity Viterbi algorithm, limit the path at any stage to m instead of fully explore the 2^l possible paths, thus dramatically reduce the complexity. In [84], it is clearly shown that the M-algorithm based multiuser detector can achieve near-optimal performance using a m value much smaller than 2^l, which is the main attraction for use in our adaptive multiuser detector structures.

4.3 Adaptive Structure

4.3.1 Receiver filter front-end

We have two choice of filter front-end: matched filter (or Ungerboeck structure) and whitened matched filter (or Forney structure).

Matched Filter

In equation 4.2, if the receiver matrix $\mathbf{g}$ is a transposed copy of the code waveform matrix $\mathbf{h}$, then the receiver is of matched version. It will result in a symmetric system matrix $\mathbf{S}$ and colored noise component $\mathbf{z}$.

$$\begin{aligned}\mathbf{y} &= \mathbf{g}_{\text{MF}}\mathbf{r} \\ &= \mathbf{h}^T\mathbf{r} = \mathbf{h}^T\mathbf{h}\mathbf{b} + \mathbf{h}^T\mathbf{n} \\ &= \mathbf{S}_{\text{MF}}\mathbf{b} + \mathbf{z}\end{aligned} \tag{4.4}$$

where $\mathbf{g}_{\text{MF}} = \mathbf{h}^T$ and $\mathbf{S}_{\text{MF}} = \mathbf{h}^T\mathbf{h}$ are the filter matrix and system matrix respectively for the matched filter structure.

Whitened Matched Filter

The noise component z is colored Gaussian noise which could be whitened. To get whitened matched filter structure, we use the Cholesky decomposition to factorize the system matrix $\mathbf{S}$:

$$\begin{aligned}\mathbf{y} &= \mathbf{S}_{\text{MF}}\mathbf{b} + \mathbf{z} \\ &= \mathbf{F}^T\mathbf{F}\mathbf{b} + \mathbf{z}\end{aligned} \tag{4.5}$$

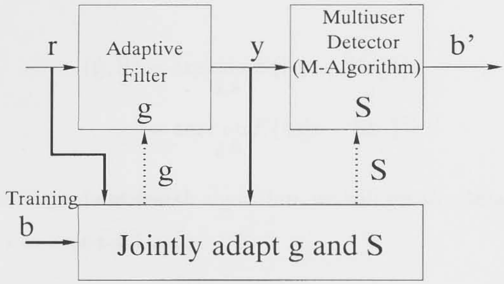


Figure 4.4: Adaptation scheme 1: joint estimation

$$
\begin{aligned}
\mathbf{y}' &= \mathbf{g}_{\mathrm{WMF}}\mathbf{r} \\
&= (\mathbf{F}^T)^{-1}\mathbf{h}^T\mathbf{r} = (\mathbf{F}^T)^{-1}\mathbf{F}^T\mathbf{F}\mathbf{b} + (\mathbf{F}^T)^{-1}\mathbf{z} \\
&= \mathbf{F}\mathbf{b} + \mathbf{n}' \\
&= \mathbf{S}_{\mathrm{WMF}}\mathbf{b} + \mathbf{n}'
\end{aligned}
\tag{4.6}
$$

where $\mathbf{g}_{\mathrm{WMF}} = (\mathbf{F}^T)^{-1}\mathbf{h}^T$ and $\mathbf{S}_{\mathrm{WMF}} = \mathbf{F}$ are the filter matrix and system matrix respectively for the whitened matched filter structure.

4.3.2 Adaptation schemes

Now let's have a look at the adaptation schemes we could explore for this multiuser detection problem.

Adapt 1: Joint adaptive scheme

The first scheme jointly adapt the receiver filter matrix $\mathbf{g}$ and system matrix $\mathbf{S}$ parameters as in Fig. 4.4, which is simpler to implement.

The filter matrix $\mathbf{g}$ and the system matrix $\mathbf{S}$ are estimated as $\hat{\mathbf{g}}$ and $\hat{\mathbf{S}}$ according to

$$
\begin{aligned}
\{\hat{\mathbf{g}}, \hat{\mathbf{S}}\} &= \arg\min_{\tilde{\mathbf{g}},\tilde{\mathbf{S}}} E\{\| \mathbf{y} - \tilde{\mathbf{S}}\mathbf{b} \|^2\} \\
&= \arg\min_{\tilde{\mathbf{g}},\tilde{\mathbf{S}}} E\{\| \tilde{\mathbf{g}}\mathbf{r} - \tilde{\mathbf{S}}\mathbf{b} \|^2\}
\end{aligned} \tag{4.7}
$$

Using stochastic gradient search algorithm, we will get the iterative adaptation steps from user i to user $i+1$ as:

$$
\begin{aligned}
\hat{\mathbf{g}}^{i+1} &= \hat{\mathbf{g}}^{i} - \mu_g \mathbf{e}_i \mathbf{r}_i \\
\hat{\mathbf{S}}^{i+1} &= \hat{\mathbf{S}}^{i} + \mu_S \mathbf{e}_i \mathbf{b}_i
\end{aligned} \tag{4.8}
$$

where $\mathbf{e}_i = \mathbf{y}_i - \hat{\mathbf{S}}^i \mathbf{b}_i$ is the estimation error, μ_g is positive step size for adapting $\hat{g}$; μ_S is positive step size for adapting $\hat{S}$.

For matched filter structure, $\hat{\mathbf{g}}$ in Eqn.4.8 is $\hat{\mathbf{g}}_{\mathrm{MF}}$ while $\hat{\mathbf{S}}$ is $\hat{\mathbf{S}}_{\mathrm{MF}}$.

For whitened matched filter structure, $\hat{\mathbf{g}}$ in Eqn.4.8 is $\hat{\mathbf{g}}_{\mathrm{WMF}}$ while $\hat{\mathbf{S}}$ is $\hat{\mathbf{S}}_{\mathrm{WMF}}$.

We can see from above that there are little computation complexity difference between the matched filter and whitened matched filter structure if using adaptive scheme 1.

Adapt 2: Channel estimation scheme

We can also estimate the user code waveforms first and then compute receiver filter matrix and system matrix from it as illustrated in Fig. 4.5.

The user spreading code matrix $\mathbf{h}$ will be estimated as $\hat{\mathbf{h}}$ according to

$$
\hat{\mathbf{h}} = \arg\min_{\tilde{\mathbf{h}}} E\{\| \mathbf{r} - \tilde{\mathbf{h}}\mathbf{b} \|^2\} \tag{4.9}
$$

The adaptive steps can be computed using gradient search method:

$$
\hat{\mathbf{h}}^{i+1} = \hat{\mathbf{h}}^{i} + \mu_h \mathbf{e}_i \mathbf{b}_i \tag{4.10}
$$

where $\mathbf{e}_i = \mathbf{r}_i - \hat{\mathbf{h}}^i \mathbf{b}_i$ is the estimation error, μ_h is positive step size for adapting $\hat{h}$.

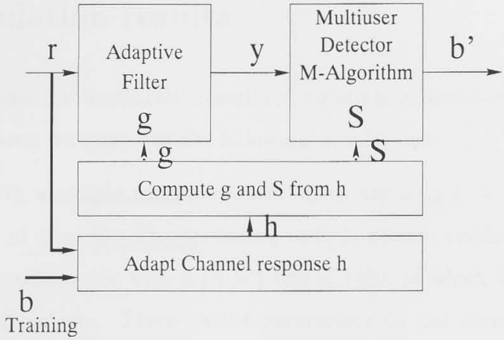


Figure 4.5: Adaptation scheme 2: channel estimation

For matched filter structure, we can derive receiver filter matrix $\hat{\mathbf{g}}_{\mathrm{MF}} = \hat{\mathbf{h}}^T$ and system matrix $\hat{\mathbf{S}}_{\mathrm{MF}} = \hat{\mathbf{h}}^T\hat{\mathbf{h}}$.

For whitened matched structure, it will take more effort. First we will get the system matrix for Matched filter structure $\hat{\mathbf{S}}_{\mathrm{MF}}$ as in previous paragraph. Then we factorize it using Cholesky decomposition and get the system matrix for whitened matched filter $\hat{\mathbf{S}}_{\mathrm{WMF}}$ and receiver filter matrix $\hat{\mathbf{g}}_{\mathrm{WMF}}$ according to Eqn.4.5 and 4.6.

There is more computational overhead for whitened matched structure compared to plain matched filter structure when using this adaptation scheme.

4.3.3 Adaptive combinations

Not surprisingly, there are a full array of combinations we could use for the adaptive multiuser detection:

$$\begin{pmatrix} \mathbf{MF} \\ \mathbf{WMF} \end{pmatrix} \begin{pmatrix} \mathbf{Adapt1} \\ \mathbf{Adapt2} \end{pmatrix} \begin{pmatrix} \mathbf{VA} \\ \mathbf{MA(m)} \end{pmatrix}$$

where MF stands for Matched filter and WMF for Whitened matched filter; VA is Viterbi algorithm and MA(m) M-algorithm (m surviving states).

4.4 Simulation results

In order to evaluate the comparative merits of various proposed adaptive multiuser detector structures, we carry out the following simulations.

We start with a simple example where there are only $l = 5$ users with a processing gain of $k = 10$. The spreading code is pseudo randomly generated. We use packet transmission with a packet size of 1400, of which 63 bits are pilot bits for training purpose. These packet parameters do not correspond directly to any current commercial CDMA systems, however, typical CDMA transmission patterns are taken into account when choosing these parameters. We use a step size $A_s = 1/l$ and $A_g = A_s/10$. The results are shown in Fig.4.6(a) and 4.6(b).

Another more complicated example simulates a lightly loaded 5 user system with spreading gain of 31. The packet size and step size are the same as previous example. The results is presented in Fig.4.7(a) and 4.7(b).

A heavily loaded system with 20 users will be our last example. Due to the exponential complexity of traditional Viterbi algorithm (in this case, in the order of 2^{20}), we focus on M-algorithm based multiuser detector. The results are shown in Fig.4.8(a) and 4.8(b).

4.5 Conclusions

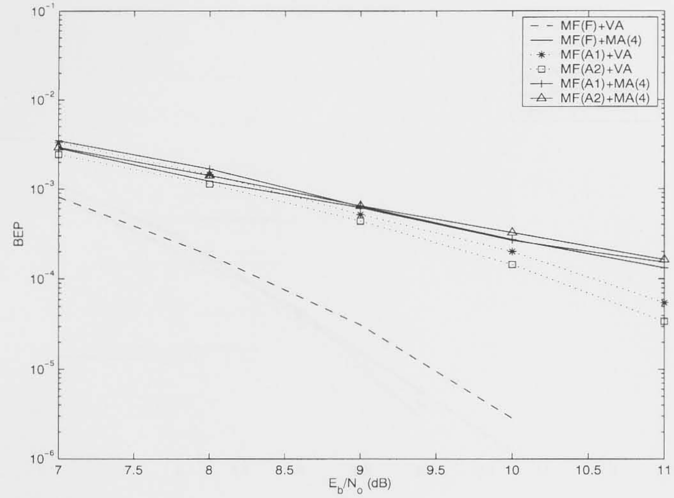
From the graphs, we can reach the following conclusions:

1) Whitened matched filter coupled with M-algorithm generally has a better bit error performance than matched filter structure. In the second example, adaptive whitened matched filter structure can achieve very good performance (close to the optimal) with a very small $m(8)$ M-algorithm, which is a significant complexity saving from 2^{20} if Viterbi algorithm is used. There is no double this kind of performance/complexity tradeoff will be very attractive for real world wireless

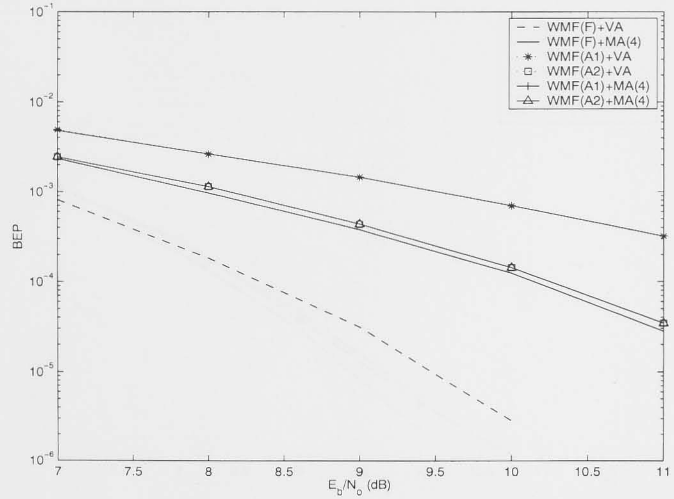
mobile CDMA communication systems.

2) At the cost of additional matrix computation, adaptive scheme 2 generally performs better than the joint adaptive scheme 1. However this involves Cholesky decomposition and matrix multiplication for each training packet. This overhead is shared by the pay load bits of the whole packet. The designer of the communication system could use the above observation as a guideline to choose their implementation depending on their specific performance/cost requirements.

3) Generally speaking, whitened matched filter front end using joint adaptive scheme and backed by m-algorithm multiuser detector is a good compromise regarding performance and complexity. By carefully selecting the step size for adaptation, convergence can be achieved by using medium training length for heavily loaded CDMA system in a typical signal to noise ratio settings.

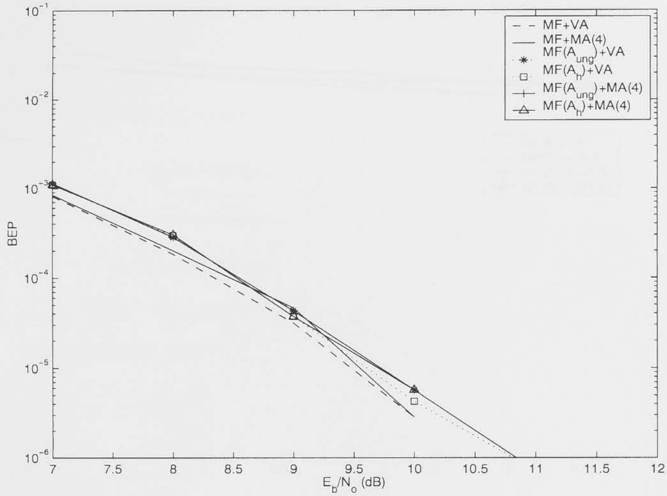


(a) Matched filter front-end

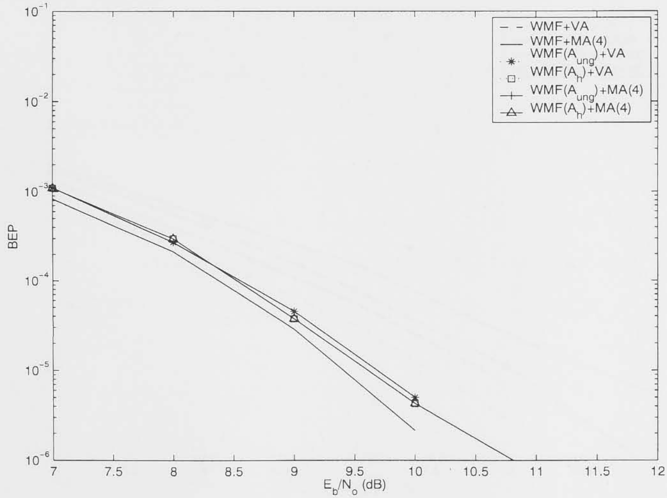


(b) Whitened matched filter front-end

Figure 4.6: Adaptive multiuser detection using different filter front-end; User:5; Spreading gain:10

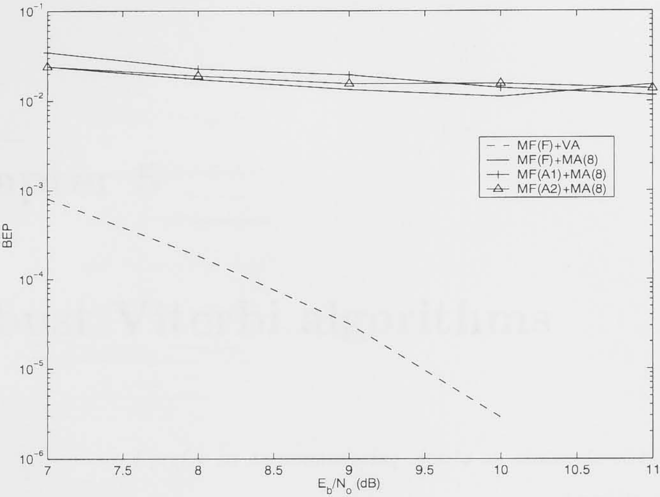


(a) Matched filter front-end

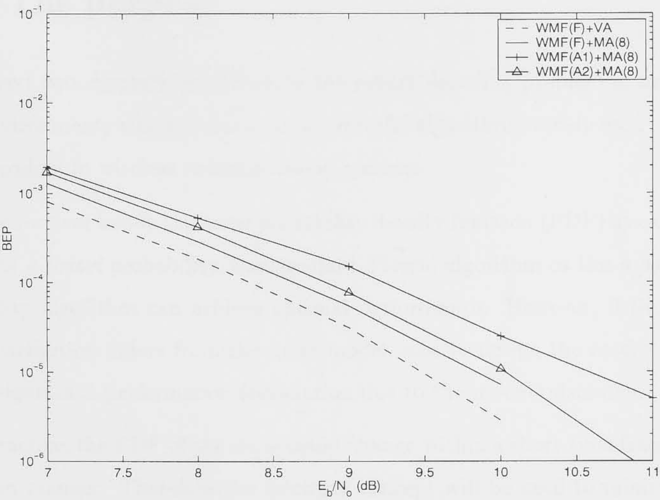


(b) Whitened matched filter front-end

Figure 4.7: Adaptive multiuser detection using different filter front-end; User:5; Spreading gain:31



(a) Matched filter front-end



(b) Whitened matched filter front-end

Figure 4.8: Adaptive multiuser detection using different filter front-end; User:20; Spreading gain:31

Chapter 5

Robust Viterbi algorithms

I prefer an accommodating vice to an obstinate virtue.
- *Amphitryon*

5.1 The purpose

In the next two chapters we will study the robust decoding problem in uncertain noise environment, with a focus on some powerful algorithms widely used for error control coding in wireless communication systems.

If the receiver knows the noise probability density function (PDF) at each time slot or its *a priori* probability, the standard Viterbi algorithm or the *a posteriori* probability algorithm can achieve optimal performance. However, if the actual noise distribution differs from the noise model used to design the receiver, there can be significant performance degradation due to the model mismatch.

In practice, the PDF of the noise could change within a short time frame in an uncertain manner. Therefore the minimax concept will be used to minimise the worst possible error performance over a family of possible channel noise PDFs.

5.2 The decoding problem

5.2.1 The system

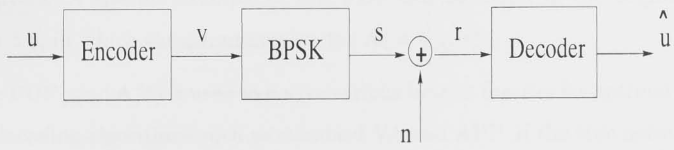


Figure 5.1: System Block Diagram for VA convolutional decoder

Consider a rate k/n convolutional code with memory length m. Let the input and output sequences of the convolutional encoder be $[\mathbf{u}]_1^l = [u_1, u_2, \ldots, u_{kl}]$ and $[\mathbf{v}]_1^l = [v_1, v_2, \ldots, v_{nl}]$ respectively, where l is the length of the sequence, $[u_{ik-k+1}, \ldots, u_{ik}]$ and $[v_{in-n+1}, \ldots, v_{in}]$ are the k-tuple of input and the n-tuple output bits at time i respectively, and $u_i, v_i \in \{0, 1\}$.

For *binary phase shift keying* (**BPSK**) modulation, we have the transmitted sequence $[\mathbf{s}]_1^l = [s_1, s_2, \ldots, s_{nl}]$, where

$$s_i = \sqrt{E_b k/n}(2v_i - 1) \tag{5.1}$$

and E_b is the signal power per source bit.

The received signal $[\mathbf{r}]_1^l = [r_1, r_2, \ldots, r_{nl}]$ after a non-fading, frequency-flat channel can be expressed as:

$$r_i = s_i + n_i, \tag{5.2}$$

where n_i is the channel noise whose PDF can change. In this thesis, we assume that the noise variables n_i for time slots $i = 1, \ldots, nl$, are mutually independent.

Assume the noise model for time slot i depends on a parameter vector $\mathbf{A}_i$ (which can indicate, e.g., noise type and noise variance). The noise PDF of the channel (denoted by α_0) and noise PDF used for decoder design (α_d) [14] is written as

$$p(n_i; \mathbf{A}_i^{\alpha_0}),$$
$$p(n_i; \mathbf{A}_i^{\alpha_d}) \tag{5.3}$$

respectively. A specific example of this PDF can be found in the beginning of Section 5.5, in which the parameter vector $\mathbf{A}_i = (\gamma_i, \sigma_i^2)$.

The PDF $p(n_i; \mathbf{A}_i^{\alpha_d})$ is used to derive various branch metrics for optimal trellis-based decoding algorithms such as standard VA and APP. If the true noise model is known, or $\mathbf{A}_i^{\alpha_d} = \mathbf{A}_i^{\alpha_0}$, then the VA and APP decoders are optimal. When the true noise model is unknown, we have $\mathbf{A}_i^{\alpha_d} \neq \mathbf{A}_i^{\alpha_0}$, and the standard VA and APP decoders are no longer optimal (they are mismatched). Therefore to achieve optimal performance for the receiver we need to know the exact noise PDF, in our case, the parameter vector α_0. In practice this is often difficult because of the existence of impulsive noise caused by either natural phenomenon such as lightening or man-made noise such as power-line noise, automobile noise, etc. The impulsive noise often lasts a short period of time, which makes noise estimation a very difficult task.

The decoder design clearly depends on how much information the receiver knows about the channel noise. In [71], Van Trees studied detecting (a) known signals in noise, b) deterministic signals with unknown parameters in noise, and (c) random signals in noise. This can be translated to noise also: (a) is akin to assuming Gaussian noise with known variance, (b) is akin to assuming the generalized noise density function but with unknown γ_i and/or σ_i^2, (c) is akin to assuming γ_i and/or σ_i^2 are random variables with some (known or measured) PDF.

The following sections will address each case respectively. Case (a) can be solved using traditional detectors, which is briefly discussed in 5.2.2 for comparison purpose. Case (c) can be solved by averaging over the parameters using their PDF, which is presented in 5.2.3. Case (b) will be the major focus of our robust detector and will be fully studied in later sections. Other techniques dealing with case (b) are mainly estimating γ_i [32] and/or σ_i^2 [68] [58].

The major difference between this work and previous works [68] [58] [32] is that we show how to design robust decoders while the above mentioned works focus on how to estimate the noise parameters. Our work can deal with situations where there are mixed noise within one data block as well as across several blocks. Thus our work and above mentioned works are complimentary. We can use the noise model estimation module for the whole block while using our robust detector to fight further uncertainties within the block.

5.2.2 Optimal decoder with perfect knowledge of channel noise

If we have perfect knowledge of the channel noise, i.e., the exact noise PDF parameter vector $\mathbf{A}_i^{\alpha_d} = \mathbf{A}_i^{\alpha_0}$ at each time slot i is known, then the optimal maximum likelihood detector (**MLD**) rule is to select the information vector $[\hat{\mathbf{u}}]_1^l$ which minimises the state metric

$$[\hat{\mathbf{u}}]_1^l = \arg\min_{[\mathbf{u}]_1^l} \{\sum_{i=1}^{nl} -\log[p(r_i - \hat{s}_i; \mathbf{A}_i^{\alpha_0})]\}. \tag{5.4}$$

The conventional Viterbi algorithm can be used to find the best path. The only modification will be replacing $|r_i - \hat{s}_i|^2$ by $-\log[p(r_i - \hat{s}_i; \mathbf{A}_i^{\alpha_d})]$ in branch metric computation. To obtain this optimal decoder, we need to estimate the PDF of the noise from bit duration to bit duration, which could be very difficult in practice.

5.2.3 Optimal decoder with knowledge of channel noise *a priori* probability

If we know the *a priori* probability of noise PDF parameter vector $\mathbf{A}_i^{\alpha_0}$, denoted as $P(\mathbf{A}_i^{\alpha_0})$, but we do not know the exact noise PDF at each time slot i, then the optimal detector can be obtained as follows.

The optimal maximum likelihood detection rule is to select the information sequence $[\hat{\mathbf{u}}]_1^l$ which minimises the metric:

$$[\hat{\mathbf{u}}]_1^l = \arg\min_{[\mathbf{u}]_1^l} \{\sum_{i=1}^{nl} -\log[p_{ave}(r_i - \hat{s}_i)]\}, \tag{5.5}$$

where

$$p_{ave}(n) = \sum_{\mathbf{A}_i^{\alpha_0}} P(\mathbf{A}_i)p(n; \mathbf{A}_i). \tag{5.6}$$

The Viterbi algorithm can be used to find the best path, with the branch metric replacing $|r_i - \hat{s}_i|^2$ by $-\log[p_{ave}(r_i - \hat{s}_i)]$. For the APP decoder [8], nothing needs to be modified except using Eqn.(5.6) to compute "$R(Y_t|X_t)$".

This decoder may be feasible, e.g., if a mobile phone can be custom made. That is, for a specific customer who is exposed to a certain pattern of man-made noise, we can measure its particular *a priori* probability $P(\mathbf{A}_i^{\alpha_0})$ and then design an VA decoder using Eqn.(5.5) and Eqn.(5.6), or an APP decoder using Eqn.(5.6). However, this is not an economical approach for mass produced handsets.

If we do not know the above *a priori* probability $P(\mathbf{A}_i^{\alpha_0})$, and use a fixed noise PDF parameter vector $\mathbf{A}_i^{\alpha_d}$ for all time slots, we end up with a mismatched decoder which could perform much worse than the matched one, as will be shown in numerical results in section 5.5. The key task of this paper is to devise a decoder to prevent significant performance loss due to the noise model mismatch. To achieve this, the "**minimax**" concept will be used.

5.3 Minimax robust decoder

5.3.1 Optimal minimax robust decoding algorithm

We now focus on case (c) (only possible types of noise are known) and first study a generalized binary hypotheses detection problem [71] using minimax techniques

[14].

Robust hypotheses detection

Suppose we have a binary hypotheses detection problem with two possible noise densities. The hypotheses are denoted as H_0 and H_1 (or -1 and 1 transmitted in Fig.5.1), while the channel noise densities $\mathbf{A}^{\alpha_0}$ can be either $\mathbf{A}^1$ or $\mathbf{A}^2$. Then each time the experiment is conducted one of eight possible things can happen:

Serial No.	Original Signal	Channel Noise	Decision
1	H_0	$\mathbf{A}^1$	H_0
2	H_0	$\mathbf{A}^1$	H_1
3	H_0	$\mathbf{A}^2$	H_0
4	H_0	$\mathbf{A}^2$	H_1
5	H_1	$\mathbf{A}^1$	H_0
6	H_1	$\mathbf{A}^1$	H_1
7	H_1	$\mathbf{A}^2$	H_0
8	H_1	$\mathbf{A}^2$	H_1

Table 5.1: Possible hypotheses detection scenario

We assign a cost coefficient C_{ki}^j to each of these occasions (H_i true, $\mathbf{A}^j$ true;

choose H_k). Then we define the risk $\Re$ to be the total expected value of cost:

$$\begin{aligned}
\Re &= \sum_{i=0}^{1}\sum_{j=1}^{2}\sum_{k=0}^{1} C_{ki}^{j} P_i P(\mathbf{A}^j) Pr(\text{choose} H_k | H_i, \mathbf{A}^j \text{true}) \\
&= \sum_{i=0}^{1}\sum_{j=1}^{2}\sum_{k=0}^{1} C_{ki}^{j} P_i P(\mathbf{A}^j) \int_{Z_k} P_{r|H_i}(\mathbf{r}|H_i, \mathbf{A}^j) d\mathbf{r} \\
&= \sum_{j=1}^{2} \Big\{ [P_0 C_{10}^j + P_1 C_{11}^j] + \int_{Z_0} \{[P_1(C_{01}^j - C_{11}^j) P_{r|H_1}(\mathbf{r}|H_1, \mathbf{A}^j)] \\
&- [P_0(C_{10}^j - C_{00}^j) P_{r|H_0}(\mathbf{r}|H_0, \mathbf{A}^j)]\} d\mathbf{r} \Big\} P(\mathbf{A}^j) \\
&= \sum_{j=1}^{2} \Re(\mathbf{A}^j) P(\mathbf{A}^j)
\end{aligned} \tag{5.7}$$

where Z_0 and Z_1 are the decision regions for hypotheses H_0 and H_1 respectively and $Z_0 \cup Z_1 = \mathbf{R}$, P_i is the probability of hypotheses H_i.

According to the minimax rule (i.e., minimise the maximum risk), we need to compute:

$$\min_{Z_0} \max_{\mathbf{A}^j} \{\Re(\mathbf{A}^j)\}. \tag{5.8}$$

For many practical cases, we can set $P_0 = P_1 = \frac{1}{2}$, $C_{10}^j = C_{01}^j = 1$ and $C_{00}^j = C_{11}^j = 0$. Then Eqn.(5.8) can be simplified as:

$$\begin{aligned}
&\min_{Z_0} \max_{\mathbf{A}^j} \left\{ \frac{1}{2} + \frac{1}{2} \int_{Z_0} [P_{r|H_1}(\mathbf{r}|H_1, \mathbf{A}^j) - P_{r|H_0}(\mathbf{r}|H_0, \mathbf{A}^j)] d\mathbf{r} \right\} \\
&= \min_{Z_0, Z_1} \max_{\mathbf{A}^j} \left\{ \frac{1}{2} \int_{Z_0} P_{r|H_1}(\mathbf{r}|H_1, \mathbf{A}^j) d\mathbf{r} + \frac{1}{2} \int_{Z_1} P_{r|H_0}(\mathbf{r}|H_0, \mathbf{A}^j) d\mathbf{r} \right\} \\
&= \min_{Z_0} \max_{\mathbf{A}^j} \{P(e|\mathbf{A}^j)\}
\end{aligned} \tag{5.9}$$

So Eqn.(5.9) is actually equivalent to minimising the maximum error probability.

Optimal robust decision rule

Clearly in Eqn.(5.9), to determine the decision rule, both minimization and maximization procedures have to be done simultaneously. In contrast, the traditional matched optimal detector is determined by a simple likelihood ratio test:

$$P_{r|H_0}(\mathbf{r}|H_0) \gtrless_{Z_1}^{Z_0} P_{r|H_1}(\mathbf{r}|H_1) \tag{5.10}$$

for every received value $\mathbf{r}$. However, the minimax procedure in Eqn.(5.9) cannot be simplified to such a decision rule which is only based on $P_{r|H_i}(\mathbf{r}|H_i, \mathbf{A}^j), i = 0, 1, j = 1, 2$ without losing its optimality. The minimax procedure is very complicated even for the binary hypotheses case as the following example shows.

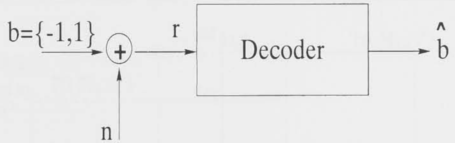


Figure 5.2: A simple detection system

Let us study this simple uncoded system (Fig.5.2), which transmits either -1 or 1 with equal-probability. The received signal is $-1 + n$ or $1 + n$, where n is noise. There are two possible types of noise in the channel, denoted by $\mathbf{A}^1$ (jagged shape in Fig 5.3) and $\mathbf{A}^2$ (step shape in Fig5.3).

In this example illustrated in Fig.5.3(a), we can see that if the received value r is within the region of $0 < r < 1$, detector 1 (matched to noise type $\mathbf{A}^1$) and detector 2 ($\mathbf{A}^2$) will give different decisions, leading to additional mismatched error. We should split this region in such a way that the detector with a larger matched error probability will have a smaller mismatched error. This will result in the minimization of maximum overall error probability (the sum of matched and mismatched error probability).

The optimal decision region Z_0 using the minimax rule (Eqn.5.9) consists of two separate subregions (Fig.5.3(b)), $r < 0.375$ and $\frac{2}{3} < r < 1$ (see Appendix I), then the error probability will be: $P(e|\mathbf{A}^1) = 0.190972 = P(e|\mathbf{A}^2)$. If the PDF of noise $\mathbf{A}^2$ becomes more complicated, then the decision region Z_0 could consist of more subregions. These subregions have little to do with the likelihood

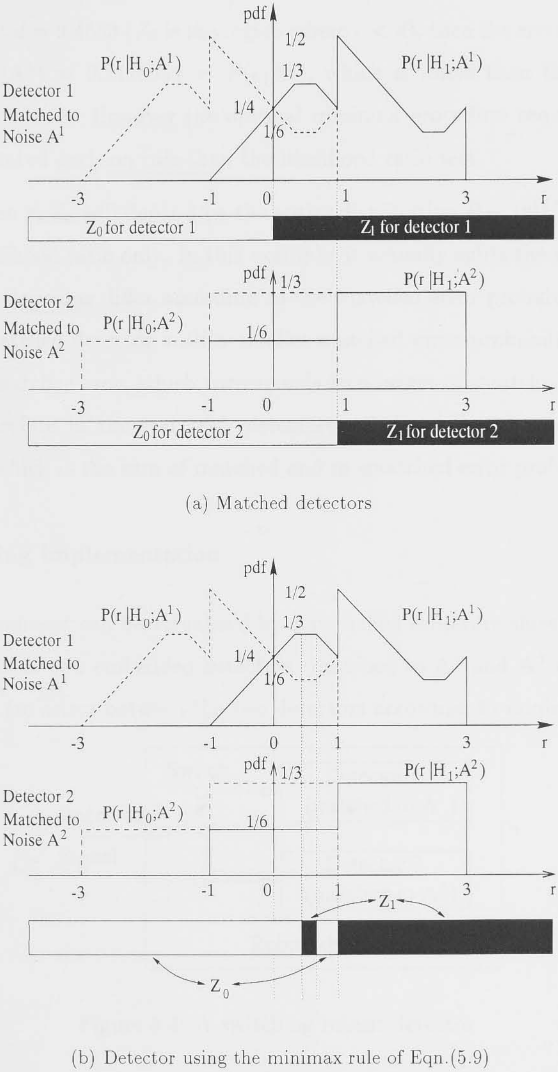


Figure 5.3: Optimal robust decoding example

ratio function at $\mathbf{r}$, but are rather determined by the factor as which will reduce the worst case error rate. If we simply use the likelihood ratio test with a single decision point $d = 0.4583$ (Z_0 is the region where $r < d$), then the error probability will be: $P(e|\mathbf{A}^1) = 0.2118054 = P(e|\mathbf{A}^2)$, which is worse than the separated decision region rule. However the optimal minimax procedure requires a much more complicated decision rule than the likelihood ratio test.

For a given $\mathbf{r}$, Z_0 will depend on the entire distribution $P_{r|H_i}(\mathbf{r}|H_i, \mathbf{A}^j)$ rather than the likelihood ratio only. In this example, it actually splits the region where the matched decisions differ according to the matched error probabilities. That is, the mismatched detector with a smaller matched error probability will have a larger mismatched area, which corresponds to a larger mismatched error probability. Therefore in the end both detectors will have the same overall error probability, which is the sum of matched and mismatched error probability.

The switching implementation

This robust concept can be visualized by a switching structure shown in Fig.5.4, which consists of two embedded detectors (matched to $\mathbf{A}^1$ and $\mathbf{A}^2$ respectively) and a switch (to select between the two detectors according to minimax rule).

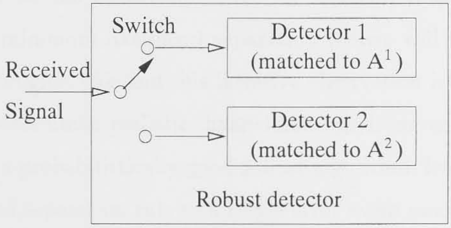


Figure 5.4: A switching robust detector

The optimal minimax switching rule can be specifies as follows:

if $r < 0$ or $r > 1$, switch to either detector;

if $0 < r < 0.375$ or $\frac{2}{3} < r < 1$, switch to detector 2;

if $0.375 < r < \frac{2}{3}$, switch to detector 1.

If we can find the optimal switching rule, then the switching detector is optimal. Otherwise, it's sub-optimal.

5.3.2 Minimax robust decoder based only on path likelihood ratio

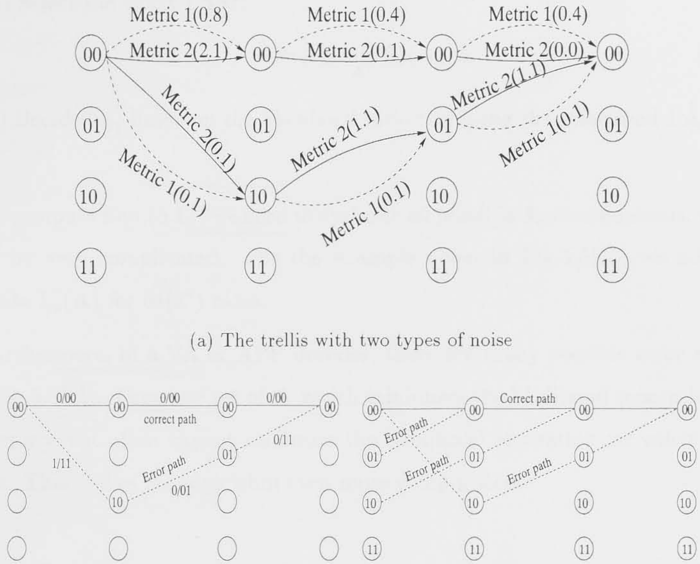
For such a simple case as in previous section, finding the switching rule according to the optimal minimax procedure is already no easy task. So for a more complicated system, can we design a simpler robust decoder which only employs the likelihood ratio information for a given r (like the optimal decoder in Eqn.(5.10))? How do we know which detector has a poorer performance based only on $P_{r|H_i}(\mathbf{r}|H_i, \mathbf{A}^j)$.

When the receiver receives r, how does it decide which detector to switch to? An intuitive approach will be switching to the detector matched to the noise model which has minimum likelihood separation metric (**LSM**), i.e.,

$$\min_{\mathbf{A}^j} \left| \log \frac{P_{r|H_0}(\mathbf{r}|H_0, \mathbf{A}^j)}{P_{r|H_1}(\mathbf{r}|H_1, \mathbf{A}^j)} \right| \tag{5.11}$$

This rule is based on the observation that the detector matched to the noise model which has minimum likelihood separation metric will be more likely to have poorer performance. We find this intuitive observation is generally true for VA and APP decoders under realistic channel noise environments. Before we dive into the details of a probabilistically good robust algorithm, let us first apply the minimum likelihood separation rule to a single error event case.

Suppose that there is a system which transmits information sequences 1 and 2 (say 000 or 100 based on a single error event of a four-state convolutional code, Fig.5.5(b)). At each transmitted bit, the channel noise $\mathbf{A}_i^{\alpha_0}$ could be either $\mathbf{A}^1$ or $\mathbf{A}^2$. We compute two branch metrics (the logarithm of the likelihood function) based on both parameter vectors. Thus we have *metric* 1 and *metric* 2 for each



(a) The trellis with two types of noise

(b) single error event

(c) Multiple error events

Figure 5.5: A system with several error paths in the trellis

branch, as the numbers in braces on dashed line and solid line in Fig.5.5(a). Note that the numbers in this graph are fictitious and only for demonstrating our path-based robust decoding concept. The received noise corrupted signal vector is $[r_1, r_2, r_3, \ldots, r_6]$. We then ask, how do we apply the minimax rule?

For this example, we first need to pinpoint the worst case. The worst case will be a possible noise pattern which has minimum separation between the likelihood of the two sequences, since it is most difficult to distinguish between the two sequences. Hence the following 3-step procedure:

(a) Compute the "*Likelihood Separation Metric*" (LSM) for each sequence

$$\mathbf{L}_p(\mathbf{A}) = \left| \log \frac{P([r]_1^l|\text{Sequence1}; \mathbf{A})}{P([r]_1^l|\text{Sequence2}; \mathbf{A})} \right| \tag{5.12}$$

for all possible parameter vector $\mathbf{A} = [\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_{nl}]$;

(b) Select the worst **LSM**:

$$\hat{\mathbf{A}} = \arg\min_{\mathbf{A}} \mathbf{L}_p(\mathbf{A}) \tag{5.13}$$

(c) Decide $[\hat{\mathbf{u}}]_1^l$ based on the likelihood ratio test using the noise model $\hat{\mathbf{A}}$ from (b).

To compute Eqn.(5.13) we need to evaluate all possible $\mathbf{A}_i$ combinations, which could be very complicated. For the example given in Fig.5.5(b), we need to compute $\mathbf{L}_p(\mathbf{A})$ for $64(2^6)$ cases.

Furthermore, in a VA or APP decoder, there are many possible error events (see Fig.5.5(c)). Thus one set of $\mathbf{A}$ which minimises the likelihood separation for one error event often cannot minimize the likelihood separation for other error events. This makes the algorithm even more complicated.

5.3.3 Minimax robust decoder based only on branch likelihood ratio

In this subsection, we will propose an sub-optimal, practically feasible minimax robust decoder based on the theories from former subsections. This is the key algorithm which will be discussed extensively.

In trellis-based decoding algorithms, if we select the minimum likelihood separation for each time instance, then it will likely minimize the likelihood separation of most error events. Based on this intuitive observation, we propose the following algorithm:

First we define the *likelihood separation metric* for each received bit:

$$\mathbf{L}_b(\mathbf{A}_i) = \left| \log \frac{p(r_i|u_i = 0; \mathbf{A}_i)}{p(r_i|u_i = 1; \mathbf{A}_i)} \right|. \tag{5.14}$$

The detector makes a decision based on the following procedure:

(a) Compute the LSM $\mathbf{L}_b(\mathbf{A}_i)$ (Eqn.5.14) for all possible parameter vectors $\mathbf{A}_i$;

(b) Find the parameter vector $\hat{\mathbf{A}}_i$ with minimal LSM for each received bit;

$$\hat{\mathbf{A}}_i = \arg\min_{\mathbf{A}_i} \mathbf{L}_b(\mathbf{A}_i); \tag{5.15}$$

(c) Use the above parameter vector to compute branch metric in standard decoding schemes. In VA, replace the branch metric computer by $-\log p(r_i|u_i = 0; \hat{\mathbf{A}}_i)$ or $-\log p(r_i|u_i = 1; \hat{\mathbf{A}}_i)$.

The important implication of (c) is that we do not need to change the trellis optimization part of traditional decoders; all we need to change is branch metric, or in most systems, a metric table in the decoder. If we know the possible noise types beforehand, this metric table can be computed off-line so there will be no additional complexity. Therefore, our minimax robust decoders will be very easy and economical to implement on top of the current decoder design.

5.4 Performance Analysis

The error performance analysis is very important in evaluating decoding algorithms. We will focus on the *bit error probability* (**BEP**) of a single error event for robust Viterbi algorithms, which is the key step to compute the Forney lower bound and union upper bounds. The strict mathematical derivation of the lower bound for a robust decoder is prohibitively difficult and will remain as an open question. Here in this section, we will numerically calculate the Forney lower bound of a minimax robust VA for a simple rate 1/2 code with 4 states and generating polynomial $[05, 07]$.

This code has a single error event with free distance $d_{\text{free}} = 5$, which is illustrated in Fig.5.6.

Because all noise density functions used in this thesis (Eqn.(5.18)) are symmetric and also the trellis is regular, we can select all zero information sequence as the transmitted sequence to calculate the error bound. Suppose the all zero

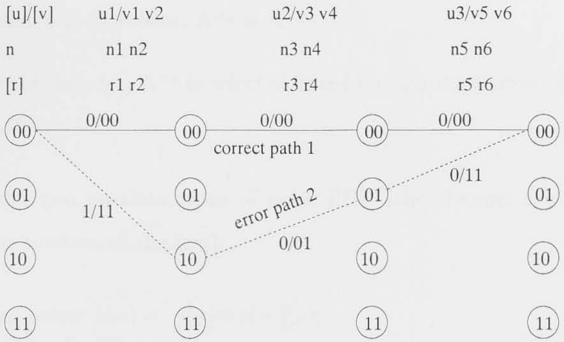


Figure 5.6: A single error event

information sequence is the solid path 1 in Fig.5.6. Path 2 (dashed line) in Fig.5.6 denotes the error path. The error event ε happens when the likelihood of path 2 is larger than that of path 1, that is:

$$\sum_{path2} bm_i > \sum_{path1} bm_i, \tag{5.16}$$

where bm_i stands for likelihood branch metric for each branch and is a function of the PDF, $p(r_i; \mathbf{A}_i^{\alpha_d})$, as given in section 5.2.

Let us define an indication function $\mathbf{I}(\mathbf{r})$ for a given $\mathbf{r}$: if the detector makes error decision, $\mathbf{I}(\mathbf{r}) = 1$; otherwise $\mathbf{I}(\mathbf{r}) = 0$.

The bit error probability on the Viterbi type decoding algorithm is:

$$P_b(e) = \int_R \omega_b \mathbf{I}(\mathbf{r}) Pr(\mathbf{r}; \mathbf{A}^{\alpha_0}) d\mathbf{r} \tag{5.17}$$

where ω_b is the Hamming weight between input bits of the two paths. In this case (Fig.5.6), $\omega_b = 1$; $Pr(\mathbf{r}; \mathbf{A}^{\alpha_0})$ is the PDF of the received signal vector $\mathbf{r} = [r_1, r_2, \ldots, r_6]$ as shown in Fig.5.6.

Now let's compute the bit error probability for three types of detectors using Eqn.(5.17):

1. The optimal matched decoder, $\mathbf{A}^{\alpha_d} = \mathbf{A}^{\alpha_0}$;

2. The mismatched decoder, $\mathbf{A}^{\alpha_d} \neq \mathbf{A}^{\alpha_0}$;

3. The robust decoder, $\mathbf{A}^{\alpha_d}$ is selected based on minimax robust procedure as Eqn.(5.15).

We consider two possible types of noise PDF (the channel could be one of them or a combination of the two):

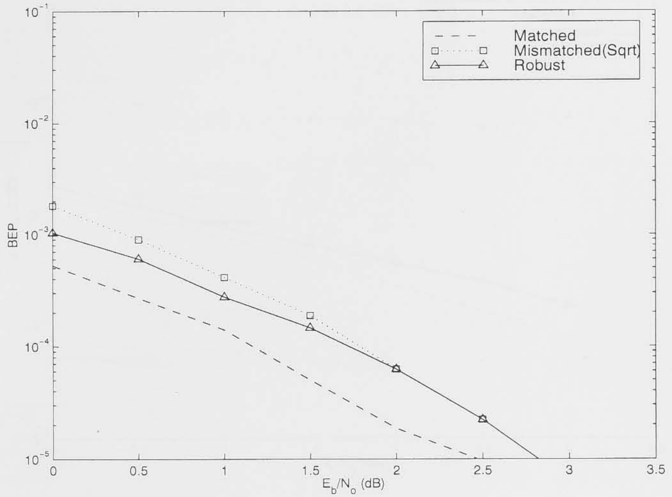
1. Gaussian noise: $p(n) = \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{n^2}{2\sigma^2})$;

2. Sqrt noise: $p(n) = \frac{1}{0.3651\sigma} exp(-\frac{\sqrt{|n|}}{0.3021\sqrt{\sigma}})$.

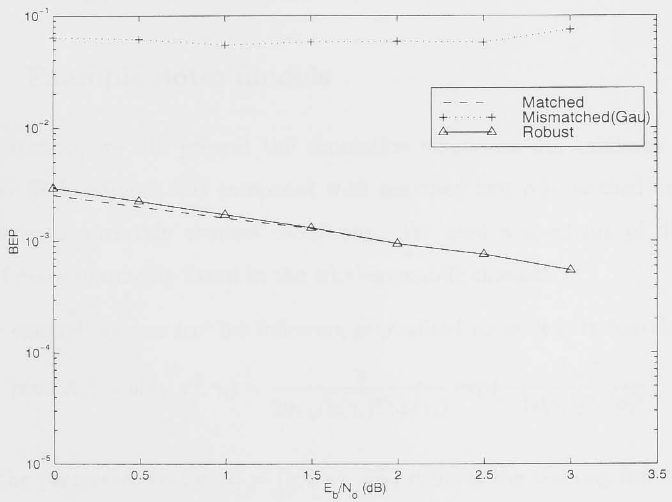
In Fig.5.7, we assume that the noise PDF is fixed for all six received bit intervals of the error event. In Fig.5.7(a) the channel noise is Gaussian noise; while in Fig.5.7(b) the channel is the Sqrt noise. In Fig.5.8, we study the case where the noise PDF at each received bit is randomly selected (with equal probability) from the Gaussian and Sqrt noise. The optimal matched receiver knows the exact noise type at each bit. The mismatched Gaussian or Sqrt receiver denotes the optimal decoder matched to either Gaussian type or Sqrt type noise only.

To compute the error rate of robust decoder, we have to average over 2^j possible noise patterns where j is the number of received bits in the error event. For this simple case, $j = 6$. Furthermore, for each noise pattern, we need to compute a j dimensional integration, which is very time consuming. It should be noted that if the channel is Gaussian, the computation of $\mathbf{I}(\mathbf{r})$ could be significantly simplified and the multiple dimensional integration could be avoided.

From Fig.5.7 and 5.8, we can see the robust decoding scheme performs better than either mismatched decoder and avoids the significant performance loss due to noise model mismatch. This will be further supported by numerical simulations in section 5.5.



(a) Gaussian Channel



(b) Sqrt Channel

Figure 5.7: Matched, mismatched and robust decoders in fixed channels

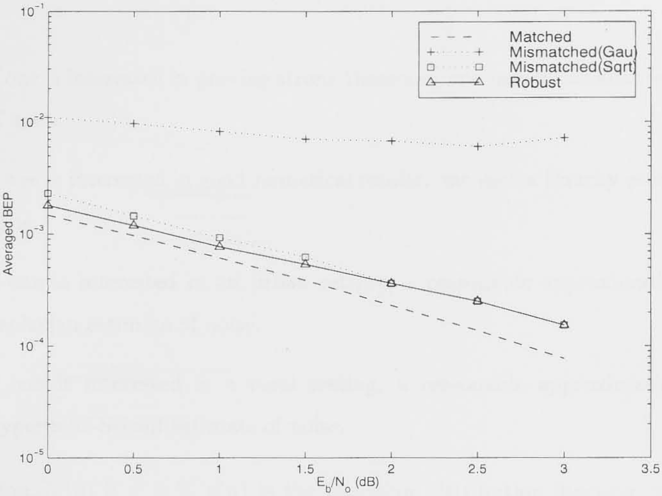


Figure 5.8: Matched, mismatched and robust decoders in random mixed channel

5.5 Simulations results

5.5.1 Example noise models

In this section, we will present the simulation results on our minimax robust decoders (Subsection 5.3.3) compared with matched and mismatched decoders under various uncertain channel conditions. We used a spectrum of different types of noise commonly found in the wireless mobile channels.

The example system has the following generalised noise density function:

$$p(n_i; \mathbf{A}_i) = p(n_i; \sigma_i^2, \gamma_i) = \frac{\gamma_i}{2\sigma_i\sqrt{a(\gamma_i)}\Gamma(1/\gamma_i)} \exp\{-\frac{|n_i|^{\gamma_i}}{[a(\gamma_i)]^{\gamma_i/2}\sigma_i^{\gamma_i}}\} \tag{5.18}$$

where the parameter vector $\mathbf{A}_i = (\sigma_i^2, \gamma_i)$, $\Gamma(\cdot)$ denotes the Gamma function, σ_i^2 is the variance of the noise and $a(\gamma_i) = \frac{\Gamma(1/\gamma_i)}{\Gamma(3/\gamma_i)}$, at time i.

This generalized noise distribution is selected trying to cover major types of real world noise. In [30] and its accompanying presentation slides, Gockenbach said:

1. If one is interested in proving strong theorems, one uses a Gaussian estimate of noise.

2. If one is interested in good numerical results, one uses a Cauchy estimate of noise.

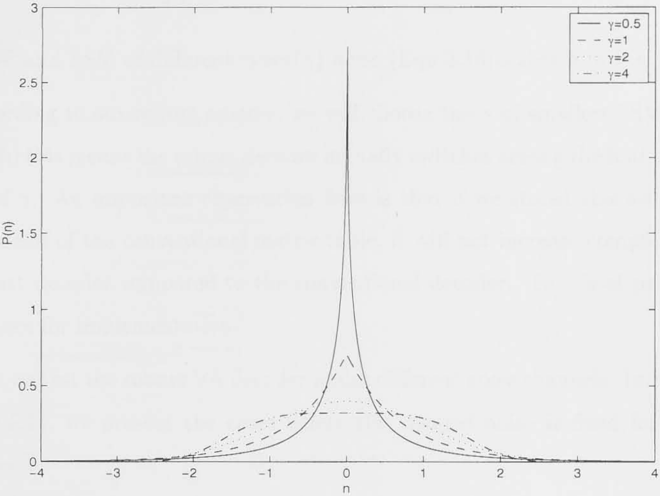
3. If one is interested in an urban setting, a reasonable approximation is a Laplacian estimate of noise.

4. If one is interested in a rural setting, a reasonable approximation is a Hyperbolic-Secant estimate of noise.

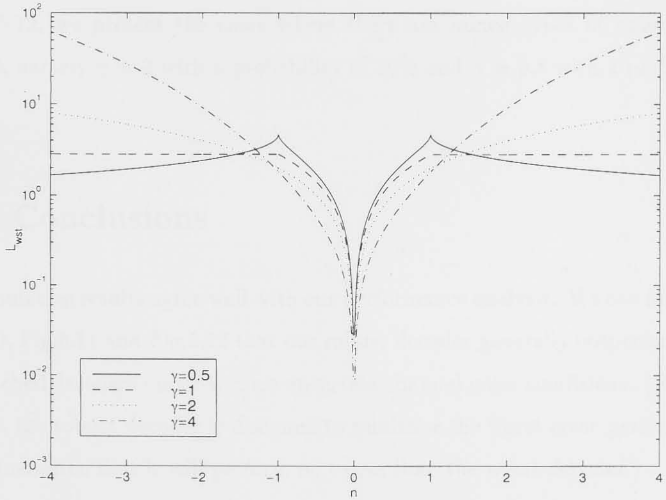
In Eqn.(5.18) if $\gamma = 2$, $p(n)$ is the Gaussian distribution function, which is important for theoretical results; if $\gamma = 1$, $p(n)$ given in Eqn.(5.18) is the Laplace distribution function, which is a good representation of the highly impulsive nature of urban airway for wireless communications; if $\gamma = 1/2$, $p(n)$ is the Sqrt noise used in Section 5.4; if $\gamma = 4$ it will be generalized Gaussian noise; if $\gamma \to \infty$ it approaches a uniform distribution. Therefore this chosen noise set is a good representative of commonly used distributions.

$$p(n_i) = \begin{cases} \frac{1}{0.3651\sigma} \exp(-\frac{\sqrt{|n_i|}}{0.3021\sqrt{\sigma}}) & \gamma = \frac{1}{2} \\ \frac{1}{\sqrt{2}\sigma} \exp(-\frac{\sqrt{2}|n_i|}{\sigma}) & \gamma = 1 \\ \frac{1}{\sqrt{(2\pi)}} \exp(-\frac{n_i^2}{2\sigma^2}) & \gamma = 2 \\ \frac{1}{3.1182\sigma} \exp(-\frac{n_i^4}{8.7539\sigma^4}) & \gamma = 4. \end{cases} \tag{5.19}$$

The VA algorithm under study is a rate 1/2 convolutional decoder with code word $[065, 057]$. All the E_b/N_o is calculated using $E_b/N_o = E_b/(2\sigma^2)$ where $\sigma^2 = 1$. We used $10,000,000$ info bits or 1000 error bits whichever reaches first. We will study how the matched decoder, mismatched decoder and our robust Viterbi algorithm decoder stack up to each other in channels with different types of noise.



(a) PDF



(b) Likelihood separation metric

Figure 5.9: Different type of noise with $\gamma = 0.5, 1, 2, 4$

5.5.2 Different types of noise

The PDF and LSM of different types(γ) noise (Eqn.5.18)is shown in Fig.5.9:

According to our robust scheme, we will choose the γ of smallest LSM. From Fig.5.9(b) this means the robust decoder actually switches among the four possible values of γ. An important observation here is that if we stored this switchable PDF instead of the conventional metric table, it will not increase complexity for the robust decoder compared to the conventional decoder. This is of particular importance for implementation.

First we test the robust VA decoder under different noise channels. In Fig.5.10 and Fig.5.11, we present the cases where the channel noise is fixed for all bit durations.

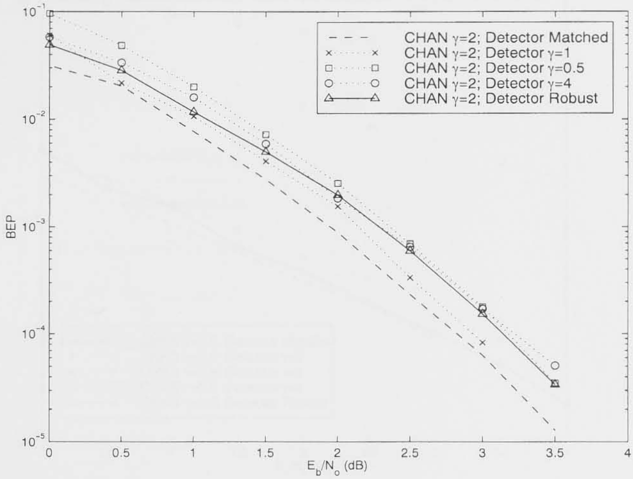
5.5.3 Complicated mixed noise

In Fig.5.12, we present the cases where there are mixed types of noise in the channel, namely $\gamma = 2$ with a probability of 80% and $\gamma = 0.5$ with a probability of 20%.

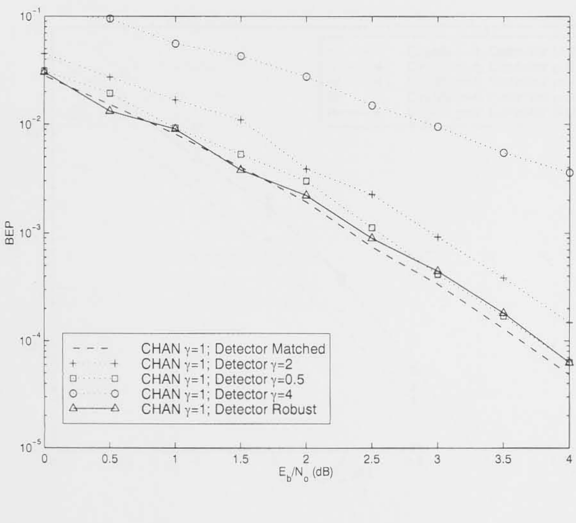
5.6 Conclusions

The simulation results agree well with our performance analysis. We can find from Fig.5.10, Fig.5.11 and Fig.5.12 that our robust decoder generally outperforms the mismatched decoders under various uncertain channel noise conditions. (Remember that the robust decoder is designed to minimise the worst error performance, which guarantee that it will perform no worse than the worst decoder.)

The figures also show that the robust decoder performs very close to the optimal matched decoder, which is a happy surprise. In fact, we have done many simulations and have not found a case where the robust VA decoder under-performs

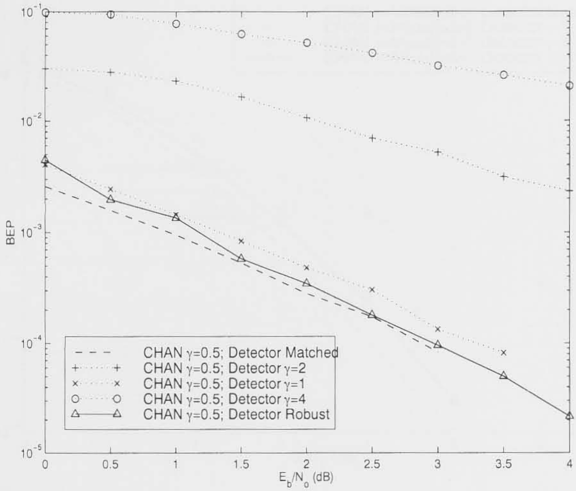


(a) Channel $\gamma = 2$

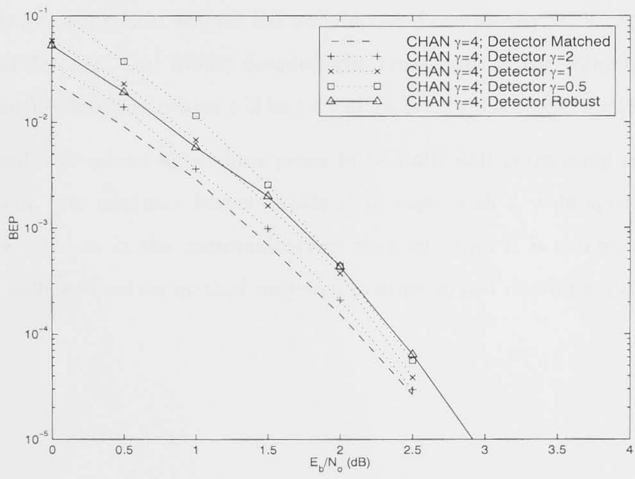


(b) Channel $\gamma = 1$

Figure 5.10: Robust VA decoder in channels with uncertain types of noise (1)



(a) Channel $\gamma = 0.5$



(b) Channel $\gamma = 4$

Figure 5.11: Robust VA decoder in channels with uncertain types of noise (2)

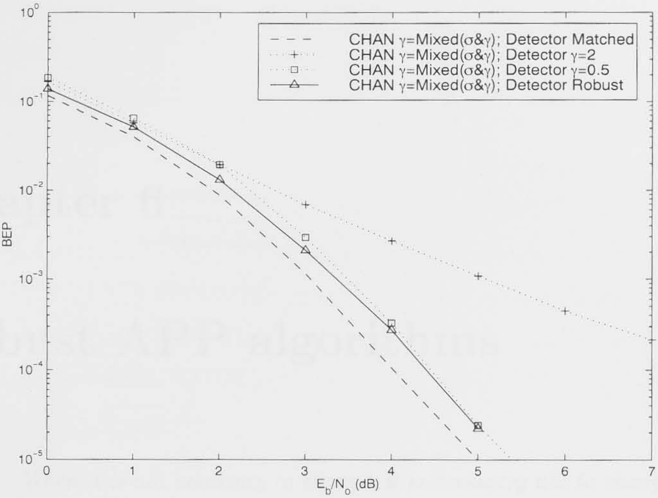


Figure 5.12: Robust VA decoder under mixed type noise channel

the optimal decoder by more than 0.5 dB.

Furthermore our robust scheme has no additional complexity for Viterbi algorithm based decoder. Our robust decoders can reuse most of the components of the standard VA decoders, which will be very attractive for industrial applications.

In a word, our robust algorithms prove to be both well performing and easy to implement. Its minimax kernel enable it to cope with a wide spectrum of uncertainty problem in the communications channel. And it is complementary with other noise estimation method proposed in other robust decoding literatures.

Chapter 6

Robust APP algorithms

When it is not necessary to change, it is necessary not to change.

- Lucius Cary

6.1 The purpose

As discussed in the Literature Review chapter, several recent publications [68] [58] [32] proposed various ways to estimate the noise variance or noise distributions for a block of transmitted signal. This chapter extends further to devise robust APP decoder based on the minimax concept for either uncertain variance, uncertain noise distribution, or a mixture of both not only within one block but also across several blocks. We will not only dealing with Turbo decoder, but also low density parity check decoder, with a possible extension to the general graph based decoding algorithms. We can also help explain some of the interesting noise variance sensitivity observations found in Summers and Reed's work.

6.2 Robust Turbo decoder

6.2.1 Turbo code

Turbo codes [11][10] is one of the most significant breakthroughs in error control coding and information theory. In Shannon's milestone paper "A Mathematical Theory of Communication", her succeeded in pointing the researchers to the ultimate goal to pursue- the Shannon limit, while omitted the systematic method, or any method at all, to achieve this limit. The road to this goal saw a lot of exciting efforts over the past 50 years, but Turbo codes clearly stands out. The original Turbo code structure is shown in Fig 6.1.

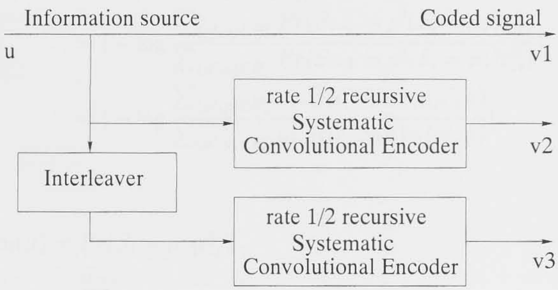


Figure 6.1: The original Turbo code

This is not only because of the excellent performance of the Turbo code , but also because of the profound conceptual implication of the coding structure and its iterative decoding scheme. The long pseudo-random interleaver shines the wisdom of information theory: randomize the code sequence; and the two simple recursive systematic convolutional code means decoding will not be prohibitively complicated. Wiberg's pioneering work to analyze the Turbo code from a general graph point of view provided us with such exciting insight as "the amazing performance of turbo code is primarily due to the cycle structure of the TWL graph..." [86].

6.2.2 Robust MAP decoder

Before we move on to derive the robust MAP Turbo decoder, let us define several notations and assumptions. Let $M(2^m)$ be the number of distinct trellis states, indexed by μ, $\mu = 0, 1, \ldots, M-1$. The state of the trellis at time i is denoted by S_i. Let B_i^0 and B_i^1 denote the sets of *transitions* $S_{i-1} = \mu' \to S_i = \mu$ that are associated with information bit $u_i = 0$ and 1 respectively.

According to our robust decoding rule,

(a) Compute LSM:

$$\begin{aligned}
\mathbf{L}(\mathbf{A}_i) &= |-\log \frac{Pr(u_i = 0|[\mathbf{r}]_1^l)}{Pr(u_i = 1|[\mathbf{r}]_1^l)}| \\
&= |-\log \frac{\sum_{(\mu',\mu)\in B_i^0} Pr(S_{i-1} = \mu'; S_i = \mu; [\mathbf{r}]_1^l)}{\sum_{(\mu',\mu)\in B_i^1} Pr(S_{i-1} = \mu'; S_i = \mu; [\mathbf{r}]_1^l)}| \\
&= |-\log \frac{\sum_{(\mu',\mu)\in B_i^0} \alpha_{i-1}(\mu')\beta_i(\mu)\eta_i(\mu',\mu)}{\sum_{(\mu',\mu)\in B_i^1} \alpha_{i-1}(\mu')\beta_i(\mu)\eta_i(\mu',\mu)}|
\end{aligned} \tag{6.1}$$

where

$$\begin{aligned}
\alpha_i(\mu) &= Pr(S_i = \mu; [\mathbf{r}]_1^i) \\
&= \sum_{\mu'=0}^{M-1} Pr(S_{i-1} = \mu'; S_i = \mu; [\mathbf{r}]_1^i) \\
&= \sum_{\mu'=0}^{M-1} Pr(S_{i-1} = \mu', [\mathbf{r}]_1^{i-1}) Pr(S_i = \mu; r_i|S_{i-1} = \mu') \\
&= \sum_{\mu'=0}^{M-1} \alpha_{i-1}(\mu')\eta_i(\mu',\mu),
\end{aligned} \tag{6.2}$$

$$\begin{aligned}
\beta_i(\mu) &= Pr([\mathbf{r}]_{i+1}^l|S_i = \mu) \\
&= \sum_{\mu'=0}^{M-1} Pr(S_{i+1} = \mu'; [\mathbf{r}]_{i+1}^l|S_i = \mu) \\
&= \sum_{\mu'=0}^{M-1} \beta_{i+1}(\mu')\eta_{i+1}(\mu',\mu),
\end{aligned} \tag{6.3}$$

$$\begin{aligned}\eta_i(\mu', \mu) &= Pr(S_i = \mu; r_i | S_{i-1} = \mu') \\ &= \sum_{\hat{s}_i} q_{1,i}(\mu|\mu') q_{2,i}(r_i|\mu, \mu') P(r_i - \hat{s}_i; \mathbf{A}_i)\end{aligned} \tag{6.4}$$

$$q_{1,i}(\mu|\mu') = \begin{cases} 1/2^k & \text{if there is a transition between } \mu' \text{ and } \mu, \\ 0 & \text{otherwise.} \end{cases} \tag{6.5}$$

suppose binary rate k/n convolutional code is used for the Turbo encoder, there will be 2^k possible transitions out of each state.

$$q_{2,i}(r_i|\mu, \mu') = \begin{cases} 1 & \text{if there is a transition between } \mu' \text{ and } \mu, \\ 0 & \text{otherwise.} \end{cases} \tag{6.6}$$

Suppose the decoder starts from state 0 and ends at state 0, then we have $\alpha_0(0) = 1, \alpha_0(\mu) = 0$ for $\mu \neq 0$; $\beta_l(0) = 1, \beta_l(\mu) = 0$ for $\mu \neq 0$.

Following the same intuitive process from optimal robust decoder to branch-based robust decoder in section 5.3, the minimization of $\mathbf{L}(\mathbf{A}_i)$ can be simplified by minimising

$$\mathbf{L}'(\mathbf{A}_i) = |-\log(\frac{P(r_i - \hat{s}_i; \mathbf{A}_i)}{P(r_i + \hat{s}_i; \mathbf{A}_i)})|. \tag{6.7}$$

(b) Choose the worst case:

$$\hat{\mathbf{A}}_i = \arg\min_{\mathbf{A}_i} \mathbf{L}'(\mathbf{A}_i). \tag{6.8}$$

(c) Compute branch metric:

Simply replacing "$R(Y_t|X_t)$" in [8] by $p_{wst}(r_i - \hat{s}_i) = p(r_i - \hat{s}_i; \hat{\mathbf{A}}_i)$ to compute the branch metrics.

6.3 Robust LDPC decoder

Low-density parity-check (LDPC) codes, proposed by Gallager in his thesis [29], is now seen as the grandfather of all of these graph codes and decoding algorithms [26]. Due to the limitation of computational power of its time, Gallager's excellent work is neglected until being re-discovered very recently.

The major attractiveness of the LDPC is its excellent performance built upon such simple structure. Gallager in his thesis provided two decoding algorithms, one simpler version which flip-flops bits until all the parity checks can be satisfied, another flooding version which is a certain variation of the sum-product algorithm.

6.3.1 LDPC code and decoder

The construction of the LDPC code is based on a very sparse parity check N by K matrix $\mathbf{G}^T$. We will give a simple example to illustrate the idea.

LDPC by example

Suppose we have a block of 4 bits to transmit, namely, s_1, s_2, s_3, s_4. For error-detection and error-correction purpose, the encoder will introduce some redundant parity check bits t_5, t_6, t_7 (notice the plus $+$ sign in the equation is the modula-2 addition):

$$\begin{aligned}
t_1 &= s_1 \\
t_2 &= s_2 \\
t_3 &= s_3 \\
t_4 &= s_4 \\
t_5 &= s_1 + s_2 + s_3 \\
t_6 &= s_1 + s_2 + s_4 \\
t_7 &= s_1 + s_3 + s_4
\end{aligned} \tag{6.9}$$

Or we can write in a matrix format, in which the generation matrix $\mathbf{G}^t$ is a $N = 7$ by $K = 4$ matrix:

$$\begin{bmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \\ t_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{bmatrix} . \tag{6.10}$$

Then we will transmit the coded sequence of $t_1, t_2, \ldots, t_7$. And the decoder can use the additional information in the parity check bits to help decoding the information bits and reduce error.

System model of LDPC

This is a very simple and powerful coding scheme. Simple as it seems to be, when the block length is long and the parity check matrix large and sparse, the LDPC is a very good code which can achieve arbitrarily small error probability at rate up to the channel capacity. [44]

To put the above coding scheme on a better mathematical ground, we can formularize it as following:

$$\begin{aligned} \mathbf{r} &= \mathbf{t} + \mathbf{n} \mod 2 \\ &= \mathbf{G}^T \mathbf{s} + \mathbf{n} \mod 2 \end{aligned} \tag{6.11}$$

In the above equations, the $\mathbf{s}$ is the uncoded information source, $\mathbf{t}$ is the coded sequence, $\mathbf{n}$ the noise in the channel and $\mathbf{r}$ the received signal.

$\mathbf{G}^T$ is the N by K dimension generator matrix and can be written in a systematic format, which consist of an $K \times K$ identity matrix part $\mathbf{I}_K$ and the parity check part $\mathbf{P}$:

$$\mathbf{G}^T = \begin{bmatrix} \mathbf{I}_K \\ \mathbf{P} \end{bmatrix} \tag{6.12}$$

The purpose of the decoding is to find the $\mathbf{s}^*$ which maximize the *a posteriori* probability:

$$\mathbf{s}^* = \arg\max_{\mathbf{s}} P(\mathbf{s}|\mathbf{r}, \mathbf{G}^T) = \arg\max_{\mathbf{s}} \frac{P(\mathbf{r}|\mathbf{s}, \mathbf{G}^T)P(\mathbf{s})}{P(\mathbf{r}|\mathbf{G}^T)} \tag{6.13}$$

LDPC Decoding algorithm

Without loss of generality, let us assume the noise $\mathbf{n}$ is independent of the information source $\mathbf{s}$ and the information source is binary equal likely. We can introduce a $N-K$ by N parity check matrix $\mathbf{A}$ which satisfies $\mathbf{A}\mathbf{G}^T = \mathbf{0}\text{mod}2$. Then apply $\mathbf{A}$ to Eqn(6.11):

$$\mathbf{A}\mathbf{n} = \mathbf{A}\mathbf{r} \mod 2 \tag{6.14}$$

Thus the decoding problem can be reduced to the task of finding the most likely noise vector $\mathbf{n}$ which satisfies:

$$\begin{aligned} &\mathbf{A}\mathbf{n} \mod 2 = \mathbf{z} \\ &\text{where} \quad \mathbf{z} = \mathbf{A}\mathbf{r} \mod 2 \end{aligned} \tag{6.15}$$

The optimal decoder, in the case of a binary symmetric channel, is an algorithm that finds the sparsest vector $\hat{\mathbf{n}}$ that satisfies $\mathbf{A}\mathbf{n} = \mathbf{z}$. Then we can obtain the transmitted signal by $\hat{\mathbf{t}} = \mathbf{r} - \hat{\mathbf{n}}$. This standardized problem can be straightforwardly solved by Belief Propagation (BP) algorithm by McKay in [44], which is actually a re-discovery of Gallager's work [29].

The algorithm comprises of four major steps: initialization, horizontal step, vertical step and decoding. Only in the first step noise related probability information is used for computing bias, all the rest three steps are iterative computations based on the bias from first step. Therefore, we'll not delve into the details of the algorithm except the "initialization" part which will be discussed in the next section.

6.3.2 Minimax robust LDPC decoder

To inject our robust kernel into the LDPC decoding algorithm, the most important change will be in the bias calculation for the initialization part. There is no need to change the horizontal step, vertical step and the decision part. More details on the algorithm can be found in [44].

The main task of the initialization is to calculate the normalized likelihood of each received bit r_l:

$$\begin{aligned} P_l^0 &= P(t_l = 0) = \frac{P(r_l|t_l=0)}{P(r_l|t_l=0)+P(r_l|t_l=1)} \\ P_l^1 &= P(t_l = 1) = \frac{P(r_l|t_l=1)}{P(r_l|t_l=0)+P(r_l|t_l=1)} \end{aligned} \tag{6.16}$$

And our robust LDPC decoding algorithm will become:

(a) Compute each received bit's likelihood separation metric for all possible parameter vectors $\mathbf{A}_l$;

$$\mathbf{L}_b(\mathbf{A}_l) = \left|\log \frac{P(r_l|t_l = 0; \mathbf{A}_l)}{P(r_l|t_l = 1; \mathbf{A}_l)}\right|. \tag{6.17}$$

(b) Find the parameter vector $\hat{\mathbf{A}}_l$ with minimal LSM for each bit;

$$\hat{\mathbf{A}}_l = \arg\min_{\mathbf{A}_l} \mathbf{L}_b(\mathbf{A}_l); \tag{6.18}$$

(c) Use the above parameter vector $\hat{\mathbf{A}}_l$ to compute the normalized likelihood bias values for each received bits using Eqn(6.16).

And the standard belief propagation network decoding algorithm will sweep through the network starting from the robust initial bias value until reach decision.

6.4 Simulations results for robust APP decoders

We will simulate our robust APP algorithm for both Turbo decoding and LDPC decoding under following uncertain channel conditions:

(a) Gaussian noise with different variance;

(b) Different types of noise;

(c) Complicated mixed noise within every block.

The tested MAP algorithm for Turbo decoding is a rate 1/2 Turbo decoder with recursive systematic code $[037, 021]$, the number of iterative decoding is 8 and the block size is 200 bits or 15000 where noted.

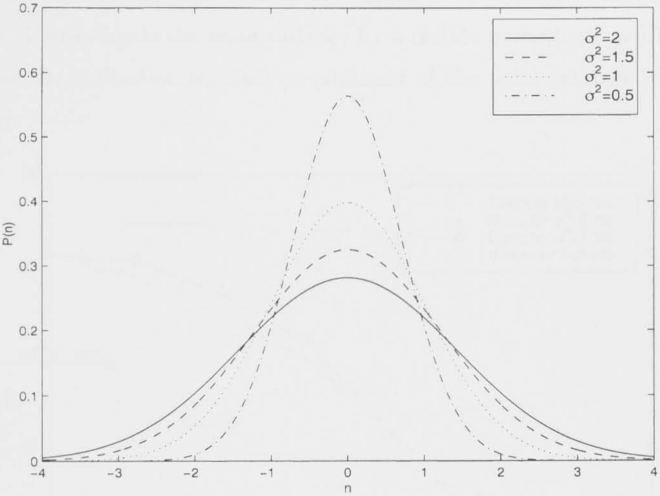
The tested APP algorithm for LDPC is a rate 1/4 parity check code with information source length $K = 3296$, parity check length $M = 10002$, packet length $N = 13298$. The maximum number of decoding iteration is 30.

All the E_b/N_o is calculated using $E_b/N_o = E_b/(2\sigma^2)$ where σ^2 is the average variance of chanel noise. For Turbo decoder, we used $15,000,000$ info bits or 1000 error bits whichever reaches first. For LDPC decoder, we used $10,000,000$ info bits or $10,000$ error bits whichever reaches first.

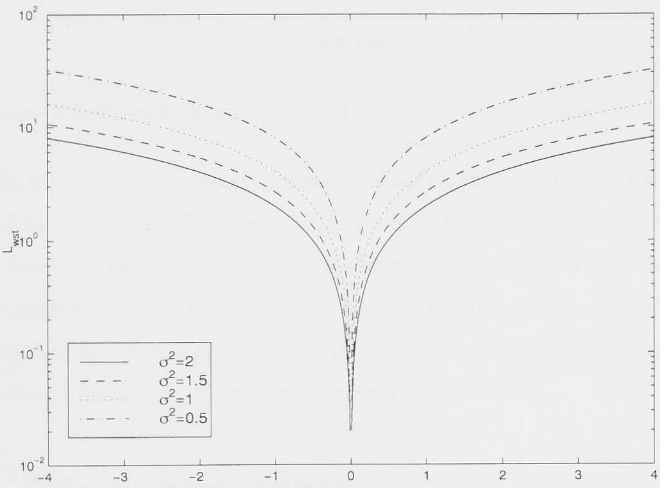
6.4.1 Gaussian noise with different noise variance

The PDF and LSM of Gaussian noise with different variances is shown in Fig.6.2:

Suppose the Gaussian noise has a variance which is only known to be within an interval $[\sigma^2_{min}, \sigma^2_{max}]$. It is well known that the VA decoder does not require the knowledge of the noise variance. However, for the MAP turbo decoder, the noise variance has to be estimated. If we could not accurately estimate the noise variance or if the received signal is affected by Gaussian noise with fluctuating variances, then according to our robust decoder rule, we should select σ^2_{max} to minimise the likelihood separation metric **LSM**. Therefore we expect that underestimating the noise variance will affect the error performance of the APP decoder



(a) PDF



(b) Likelihood separation metric

Figure 6.2: Gaussian noise with $\sigma^2 = 2, 1.5, 1, 0.5$

much more significantly than over-estimating the noise variance. Thus, a simple rule (i.e., over-estimate the noise variance by a certain percentage) can be used to reduce the estimation accuracy requirement of the noise variance in Turbo decoding.

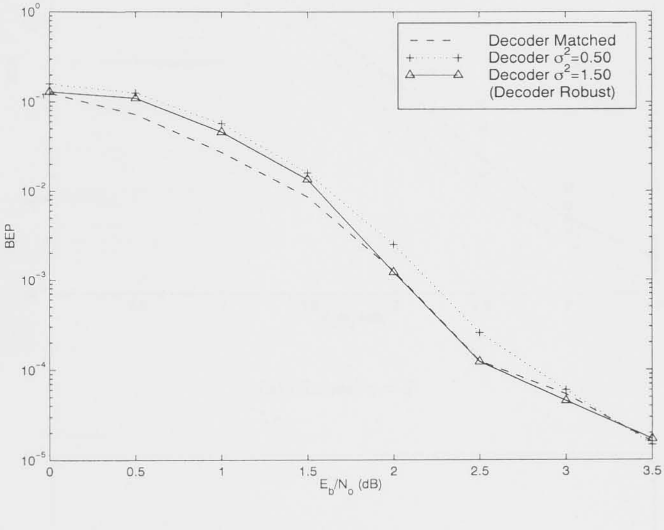
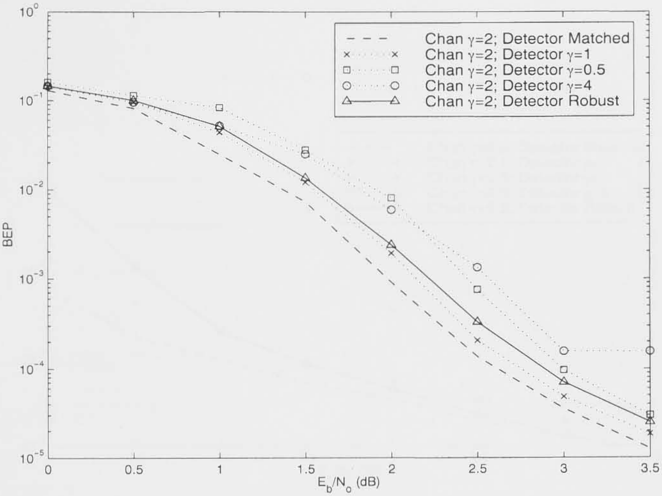


Figure 6.3: Robust Turbo decoder in uncertain variance noise channel

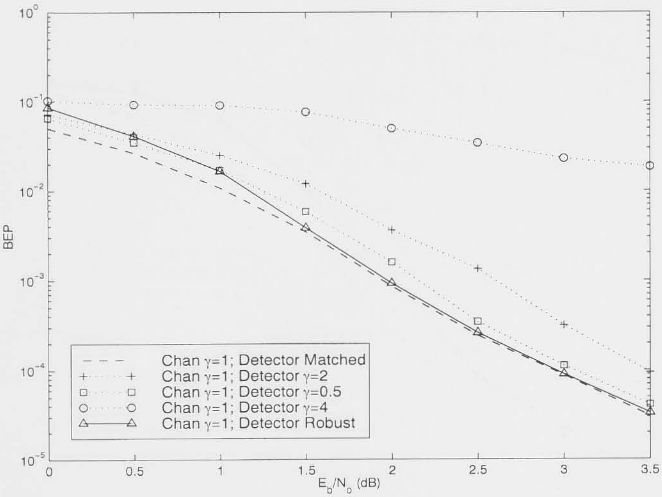
In Fig.6.3, we show the effect of noise variance estimation error on the performance of the Turbo decoder where the APP algorithm is used. From the results we find that it is better to over-estimate the noise variance by 50% rather than underestimate it by 50%. And such noise variance sensitivity behaviour is also reported in Fig.1 of [58] and Fig.2 of [68]. Our robust decoder based on minimizing the Likelihood Separation Metric helps explain this behaviour.

6.4.2 Different types of noise

Simulations are carried out for the Turbo decoder using MAP algorithm for different types of noise ($\gamma = 2$, $\gamma = 1$, $\gamma = 0.5$, $\gamma = 4$) as in Fig.6.4, Fig.6.5.

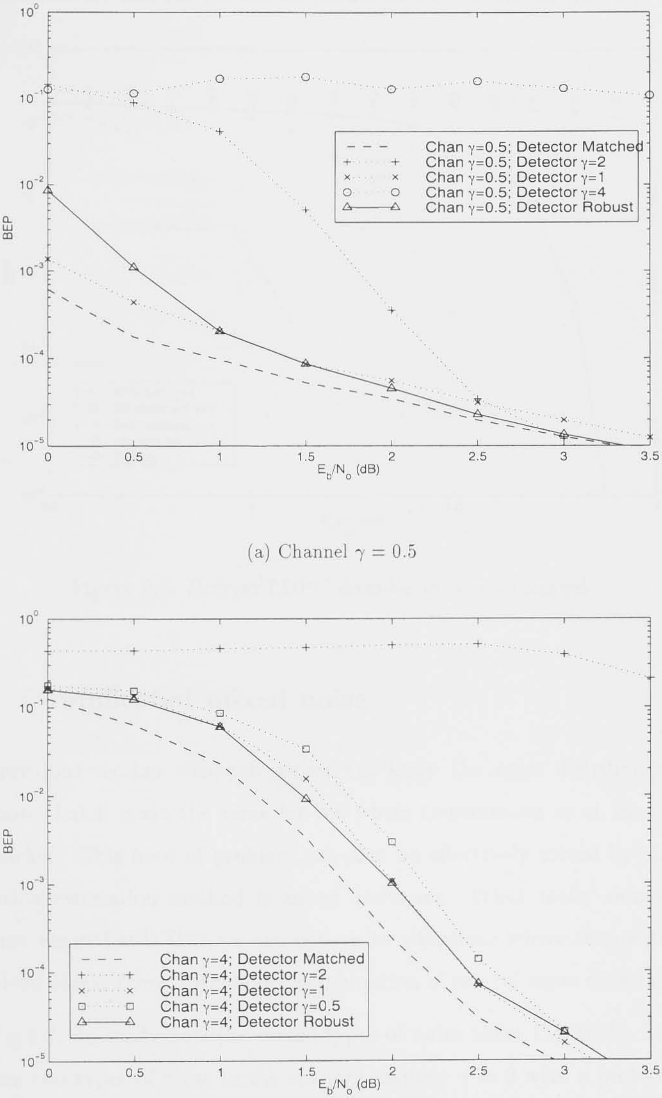


(a) Channel $\gamma = 2$



(b) Channel $\gamma = 1$

Figure 6.4: Robust Turbo decoder in different types of noise channel(1)



(a) Channel $\gamma = 0.5$

(b) Channel $\gamma = 4$

Figure 6.5: Robust Turbo decoder in different types of noise channel (2)

Simulations are also carried out for robust LDPC decoder as shown in Fig.6.6.

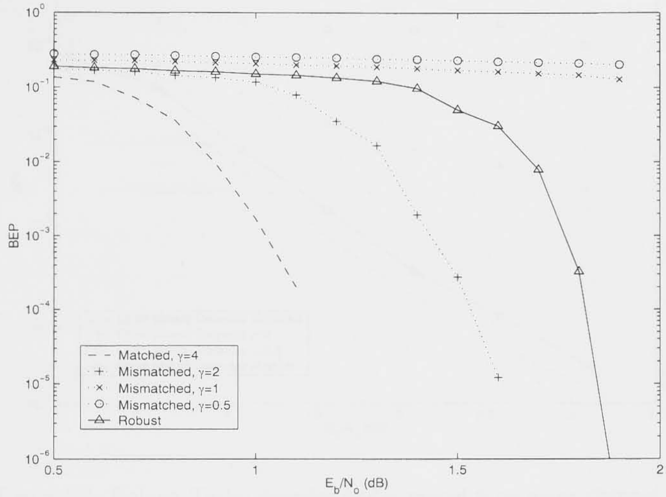


Figure 6.6: Robust LDPC decoder in $\gamma = 4$ channel

6.4.3 Complicated mixed noise

In the previous section although we do not know the noise distribution type, we suppose that it stays the same for the whole transmission or at least for the whole packet. This kind of problem can also be effectively solved by the noise distribution estimation method in other literature. What really shines about our robust algorithm is that we can cope with situations where even within one transmitted block, there are random combination of several noise distributions.

In Fig.6.7, we study how the mixed types of noise affect the Turbo decoders. There are two types of noise in the channel, namely $\gamma = 2$ with a probability of 80% and $\gamma = 0.5$ with a probability of 20%. To make this case more complicated, the $\gamma = 0.5$ type of noise has an uncertain variance between $\sigma^1 = 1$ to $\sigma^1 = 2$ equal-likely.

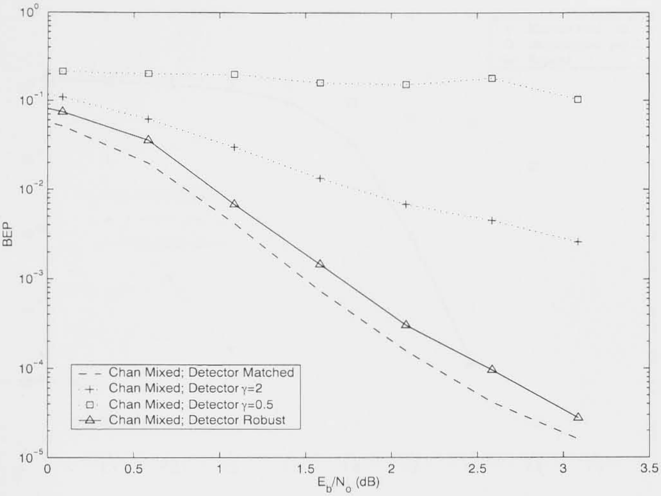


Figure 6.7: Robust Turbo decoder under mixed type noise channel 1

We tested the same mixed noise channel for LDPC decoder in Fig.6.6.

Other mixed channel situations are also studied in Fig.6.9 ($Pr(\gamma = 2) = 50\%, Pr(\gamma = 1) = 50\%(\sigma^2 = 1$ or 2 equal-likely)) and Fig.6.10 ($Pr(\gamma = 2) = 50\%, Pr(\gamma = 0.5) = 25\%, Pr(\gamma = 4) = 25\%$). They study the same Turbo decoder except this time the transmitted packets are longer at 15000 bits per packet.

6.5 Conclusions

The most important conclusion from these simulation results is that our robust APP algorithms can successfully handle various mixed noise channels, which is not being solved by any other robust decoding literature. Our robust algorithms can potentially deal with any combination of noise type and noise variance within one transmitted packet as well as across several packets.

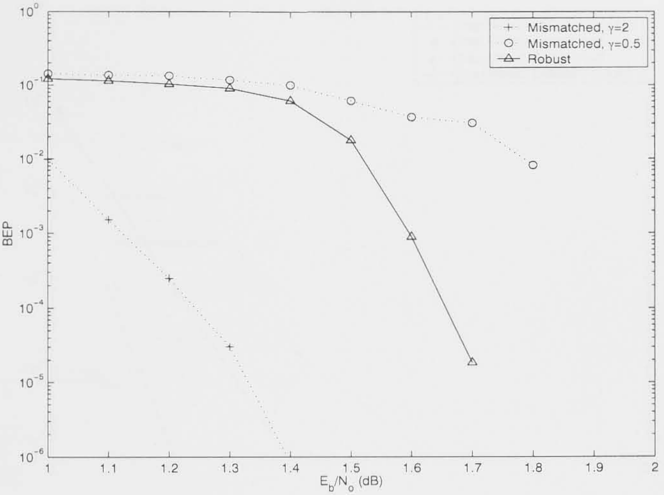


Figure 6.8: Robust LDPC decoder under mixed type noise channel 1

Our robust Turbo decoder yield strong performance results. The figures show that the behaviour of the robust Turbo decoder is similar to the robust VA decoder. However, the mismatched Turbo decoders often perform much worse than the optimal matched Turbo decoder and the robust Turbo decoder. Therefore, we believe that robust Turbo decoder is particularly useful for minimizing the performance loss due to mismatch between the design noise model and the channel noise model.

For LDPC code, our robust decoder also showed good results compared with various mismatched decoder, but it's not as good as the robust Turbo decoder because there is some noticeable performance loss compared with optimal matched LDPC decoder. So there is still lots of room for improvement.

Implementation side, the robust Turbo decoder has slight overhead to store the noise information due to the interleaver structure (the types of noise $\mathbf{A}$ for each received bit needed to be stored during the first iteration so that they don't need to be calculated again for subsequent iterations). Both robust Turbo decoder

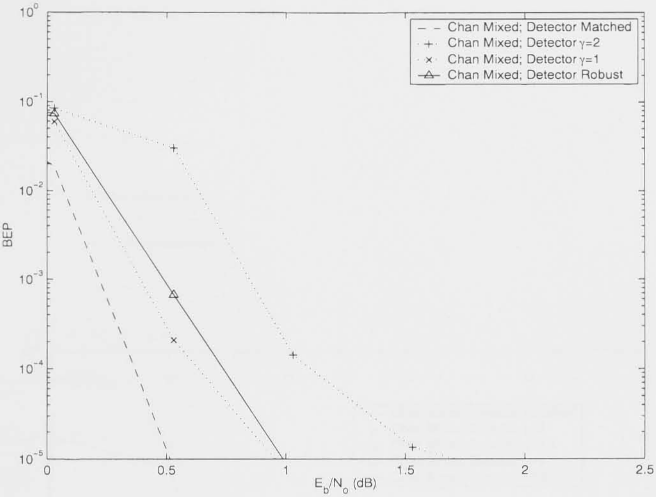


Figure 6.9: Robust Turbo decoder under mixed type noise channel 2

and robust LDPC decoder change the likelihood computation part of the standard algorithm, which could be pre-computed and stored in the likelihood table.

In conclusion, our robust APP algorithms are powerful ideas in simple format. Its potential use in mixed noise channels seems very promising.

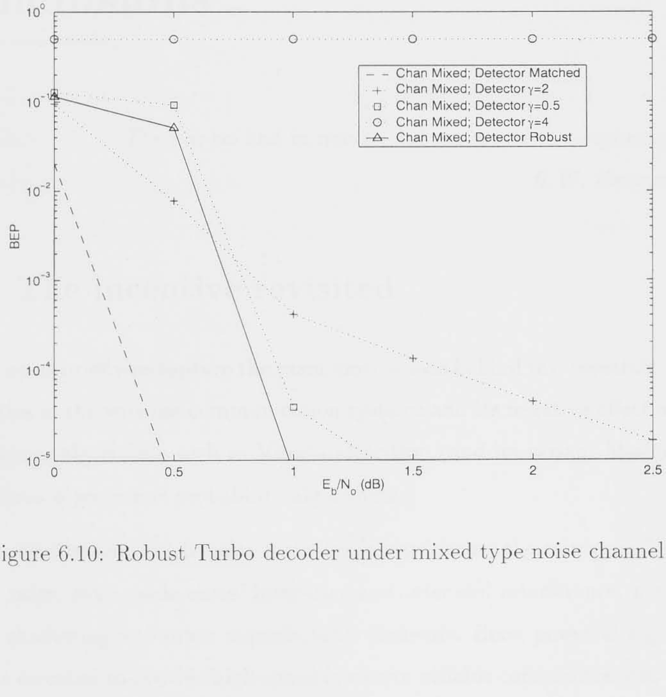


Figure 6.10: Robust Turbo decoder under mixed type noise channel 3

Chapter 7

Conclusions

There is no end in nature, but every end is a beginning.
R.W. Emerson

7.1 The incentive revisited

Firstly let us briefly re-capture the main motivations behind this research: the uncertainties in the wireless communication systems and its negative effect on high-performance algorithms such as Viterbi algorithm (and its variant M-algorithm) and various *a posteriori* probability algorithms.

The wireless channel is quite a challenge considering the mixture of environmental noise, man-made noise, inter-user and inter-cell interference, multi-path fading, shadowing and other unpredictable elements. Here, powerful algorithms, which is essential to provide high-speed low-error reliable communication, will not be able to deliver their theoretically performance or they might not work at all. Therefore both adaptive and robust approaches are used to solve the problem. The next section is a summary of our achievements.

7.2 Summary of our Achievements

We proposed new adaptive structures for Viterbi algorithm and M-algorithm backed systems, which provide adaptive parameter estimation, near-optimal performance and low implementation complexity for inter-symbol interference channel in Global System for Mobile Phones(GSM) system and multiuser detection in Code Division Multiple Access (CDMA) mobile systems.

We initiated the novel concept of minimax decoding based on the optimal robust algorithms and applied it to the traditional Viterbi algorithm and various *a posteriori* probability decoders to combat man-made noise and other uncertainties in wireless channels. Our algorithm's unique ability to cope with mixed noise problem within one transmitted packet is the first reported in literature. Our robust structure is not only easy to implement on top of existing decoders, but also provide good performance which is supported by strong simulation and analytical results.

7.3 Outlook for future works

There are lots of possible ways to extend the work presented in this thesis, either along the adaptive thread or the robust thread. There are also interesting possibilities of combining these two techniques together.

Along the adaptive path, open questions to be answered:

- The convergence issue of the various adaptive schemes proposed in Chapter 3 and 4, especially in relation to the eigen values of the system matrix and filter matrix, adaptation step size, training length and training pattern. This will provide more practical benchmark in helping designing adaptive receivers.
- Introduction of orthogonality constraint (to whiten the noise) for the joint

adaptation scheme coupled with whitened matched filter. This might lead to an algorithm which shares the simplicity of joint adaptation while providing better performance.

More research work on the robust front include:

- Generalization of the minimax robust kernel from binary hypothesis to multiple hypothesis detection. This will require a generalized definition of the likelihood separation metric, generalized hypothesis selection criteria and switching procedure.

- Application of minimax robust techniques in trellis-coded modulation, iterative Viterbi algorithm and general graph based decoding algorithms.

  Solid field test of robust enhanced receiver in commercial mobile communication systems to see the performance of our algorithm in real world conditions.

It will also be of interest to explore the possibility of marrying the two techniques together to devise some form of "adaptive - robust" or "robust - adaptive" receiver. One tentative scheme will be an adaptive filter front-end backed by a robust decoder. The adaptive filter could converge to within a certain range from the true system model. This range, instead of the exact estimated values, could be fed into the robust decoder to help decode the received information. Such an arrangement will significantly relax the estimation accuracy requirements, thus reducing estimation sensitivity and estimation complexity.

Appendix A

The Calculation of optimal minimax decision region

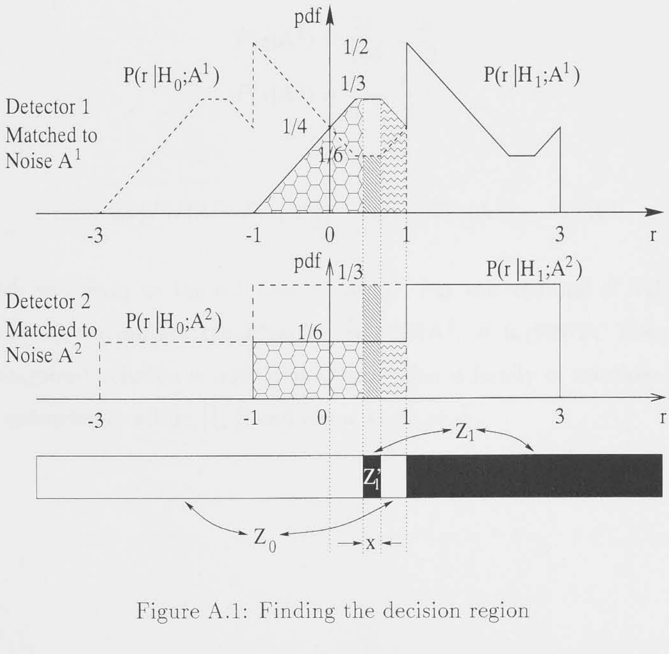


Figure A.1: Finding the decision region

In Fig.A.1, the r axis is divided into three regions, namely $r \leq 0$, $0 < r \leq 1$ and $r > 1$. For the first and third regions, the decision will be the same no matter which detector is used. For the middle region ($0 < r \leq 1$), conflicting decisions will be made by different detectors. The optimal minimax robust decoder will split this region in such a way as to minimize the maximum error probability.

Suppose we will partition the region $0 < r \leq 1$ into Z_0' and Z_1'. Let the area Z_1' has a width of x (unknown at this stage), where should we put this subregion? No matter where we put this subregion Z_1', the mismatched PDF will be the same ($\frac{1}{3} - \frac{1}{6} = \frac{1}{6}$) for detector 2 (Matched to A^2). But for detector 1 (matched to A^1), if we put the subregion Z_1' between $\frac{1}{3}$ and $\frac{2}{3}$, then we can avoid the biggest mismatched error PDF ($\frac{1}{3} - \frac{1}{6} = \frac{1}{6}$). The error probability can be calculated by adding together the three shaded areas (honeycomb, hash-line and wave-line) in Fig.A.1:

$$\begin{aligned} P(e|\mathbf{A}^1) &= \frac{31}{144} - \frac{x}{12} \\ P(e|\mathbf{A}^2) &= \frac{1}{6} + \frac{x}{12}. \end{aligned} \tag{A.1}$$

$$\min\max\{P(e|\mathbf{A}^1), P(e|\mathbf{A}^2)\} \quad \longrightarrow \quad P(e|\mathbf{A}^1) = P(e|\mathbf{A}^2) \tag{A.2}$$

which will gives us the solution $x^* = \frac{7}{24}$. Put this optimal x^* value back in Eqn.(A.1), we can get the $P^*(e|\mathbf{A}^1) = P^*(e|\mathbf{A}^2) = 0.190972$. The optimal minimax robust solution is not unique, but rather a family of solutions as long the Z_1' subregion is within $[\frac{1}{3}, \frac{2}{3}]$ and has a width of $\frac{7}{24}$.

Bibliography

[1] B. Aazhang and H.V. Poor. Performance of DS/CDMA communications in impulsive channels -Part I: Linear correlation receivers. *IEEE Transaction on Communications*, 35(11):1179–1187, 1987.

[2] K. Abend and B.D. Fritchman. Statistical detection for communication channels with intersymbol interference. *Proceedings of IEEE*, 58(5):779–785, 1970.

[3] K. Abend, T.J. Hartley, B.D. Fritchman, and C. Gumacos. On optimal receivers for channels having memory. *IEEE Transaction on Information Theory*, 14(11):152–157, 1968.

[4] J.B. Anderson. Limited search trellis decoding of convolutional codes. *IEEE Transaction on Information Theory*, 35(5):944–955, 1989.

[5] J.B. Anderson. Reduced-state sequence detection with convolutional codes. *IEEE Transaction on Information Theory*, 40(3):965–972, 1994.

[6] J.B Anderson and S.M. Hladik. Tail-biting MAP decoders. *IEEE Journal on Selected Areas in Communications*, 16:297–302, 1998.

[7] T. Aulin. Optimal trellis decoding at given complexity. In *IEEE Int. Symp. on Information Theory*, pages 17–22, San Antonio, Texas, USA, 1993.

[8] L.R. Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Transaction on Information Theory*, 20(2):284–287, 1974.

[9] C.T. Beare. The choice of the desired impulse response in combined linear-Viterbi algorithm equalisers. *IEEE Transaction on Communications*, 26(8):1301–1307, 1978.

[10] C. Berrou and A. Glavieux. Near optimum error correcting coding and decoding: Turbo-codes. *IEEE Transaction on Communications*, 44(10):1261–1271, 1996.

[11] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon limit error-correcting coding and decoding: Turbo codes(1). In *Proc. of ICC'93*, pages 1064–1070, Geneva, Switzerland, 1993.

[12] K.L. Blackard, T.S. Rappaport, and C.W. Bostian. Measurements and models of radio frequency impulsive noise for indoor wireless communications. *IEEE Journal on Selected Areas in Communications*, 11(7):991–1001, 1993.

[13] T.K. Blankenship, D.M. Krizman, and T.S. Rappaport. Measurements and simulation of radio frequency impulsive noise in hospitals and clinics. In *Proc. 1997 IEEE Vehicular Technology Conference (VTC'97)*, pages 1942–1946, 1997.

[14] R.K. Boel, M.R. James, and Petersen I.R. Robustness and risk-sensitive filtering. *Submitted for publication in IEEE Trans. Automatic Control*, 1998.

[15] A. Duel-Hallen. Decision-feedback multiuser detector for synchronous code-division multiple-access channel. *IEEE Transaction on Communications*, 41(2):285–290, 1993.

[16] A. Duel-Hallen and C. Heegard. Delayed decision-feedback sequence estimation. *IEEE Transaction on Communications*, 37(5):428–436, 1989.

[17] A. Dutta-Roy. Bring home the Internet. *IEEE Communication Magazine*, 36(3):33–38, 1999.

[18] A. Dutta-Roy. Cable: It's not just for TV. *IEEE Communication Magazine*, 36(5):53–59, 1999.

[19] M.V. Eyuboglu and S.U.H. Qureshi. Reduced-state sequence estimation with set partitioning and decision feedback. *IEEE Transaction on Communications*, 36(1):13–20, 1988.

[20] M.V. Eyuboglu and S.U.H. Qureshi. Reduced-state sequence estimation for coded modulation on intersymbol interference channels. *IEEE Journal on Selected Areas in Communications*, 7(6):989–995, 1989.

[21] D.D. Falconer and F.R. Magee. Adaptive channel memory truncation for maximum-likelihood sequence estimtion. *Bell Syst. Tech. J.*, 52:1541–1562, 1973.

[22] B. Farhang-Boroujeny. Channel estimation via channel identification: Algorithms and simulation results for rapidly fading HF channels. *IEEE Transaction on Communications*, 44(11):1409–1412, 1996.

[23] J. Farrel and M.L. Katz. Public policy and private investment in advanced telecommunications infrastructure. *IEEE Communication Magazine*, 36(7):87–92, 1998.

[24] G.D. Jr. Forney. Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference. *IEEE Transaction on Information Theory*, 18(3):363–378, 1972.

[25] G.D. Jr. Forney. On iterative decoding and the two-way algorithm. In *Intl. Symp. Turbo Codes and Related Topics*, Brest, France, 1997.

[26] G.D. Jr. Forney, F.R. Kschischang, and B. Marcus. Iterative decoding of tail-biting trellis. In *Information Theory Workshop*, San Diego, CA, USA, 1998.

[27] J.G. Proakis F.R. Magee Jr. Adaptive maximum-likelihood sequence estimation for digital signalling in the presence of intersymbol interference. *IEEE Transaction on Information Theory*, 19:120–124, 1973.

[28] R.G. Gallager. Low-density parity-check codes. *IRE Trans. Information Theory*, 8:21–28, 1962.

[29] R.G. Gallager. *Low-Density Parity-Check Codes*. MIT Press, Cambridge, MA, USA, 1963.

[30] M.S. Gockenbach and A.J. Kearsley. Optimal Signal Sets for Non-Gausian Detectors. *SIAM Journal of Optimization*, 9(2):316–326, 1999.

[31] Y. Gu and T. Le-Ngoc. Adaptive combined DFE/MLSE techniques for ISI channels. *IEEE Transaction on Communications*, 44(7):847–856, 1996.

[32] X.L. Huang and Nam Phamdo. Turbo decoders which adapt to noise distribution mismatch. *IEEE Communication Letters*, 2(12):321–323, 1998.

[33] B.E. Keiser and E. Strange. *Digital Telephony and Network Integration*. International Thomas Publishing Inc., 1995.

[34] T. Koizumi, Y. Inoue, and M. Ohta. The effect of non-Gaussian noise on the performance of binary CPSK systems. *IEEE Transaction on Communications*, 26(2):305–309, 1978.

[35] L. Lange. The Internet. *IEEE Spectrum*, 36(1):35–40, 1999.

[36] G. Lawton. Technologies take the network home. *IEEE Computer*, 32(3):11–14, 1999.

[37] W.C.Y. Lee. *Mobile Communications Design Fundamentals*. John Wiley and Sons Inc., USA, 1993.

[38] W.U. Lee and F.S. Hill. A maximum-likelihood sequence estimator with decision-feedback equalization. *IEEE Transaction on Communications*, 25(9):971–979, 1977.

[39] R. Lupas and S. Verdu. Linear multiuser detectors for synchronous code-division multiple-access channels. *IEEE Transaction on Information Theory*, 35(1):123–136, 1989.

[40] R. Lupas and S. Verdu. Near-far resistance of multiuser detection in asynchronous channels. *IEEE Transaction on Information Theory*, 38(3), 1990.

[41] U. Madhow and M.L. Honig. MMSE interference suppression for direct-sequence spread-spectrum CDMA. *IEEE Transaction on Communications*, 42(12):3178–3188, 1994.

[42] P.L. McAdam, L. Welsh, and Weber C. MAP bit decoding of convolutional codes. In *IEEE Int. Symp. on Information Theory*, Asilomar, California, USA, 1972.

[43] R.J. McEliece, D.J.C. MacKay, and J.-F. Cheng. Turbo decoding as an instance of Pearl's belief propagation algorithm. *IEEE Journal on Selected Areas in Communications*, 16:140–152, 1998.

[44] D.J.C. McKay. Good eror-correcting codes based on very sparse matrices. *IEEE Transaction on Information Theory*, 1997.

[45] D. Middleton. Man-made noise in urban environments and transportation systems: Models and measurement. *IEEE Transaction on Communications*, 21:1232–1241, 1973.

[46] D. Middleton. Statistical-physical models of electromagnetic interference. *IEEE Trans. Electromag. Compat.*, 19:106–127, 1977.

[47] J.H. Miller and J.B. Thomas. Detectors for discrete-time signals in non-Gaussian noise. *IEEE Transaction on Information Theory*, 18(3):241–250, 1972.

[48] J.W. Modestino and A.Y. Ningo. Detection of weak signals in narrowband non-Gaussian noise. *IEEE Transaction on Information Theory*, 25(9):552–560, 1979.

[49] S. Morgan. The internet and local telephone network: Conflicts and opportunities. *IEEE Communication Magazine*, 36(1):42–48, 1998.

[50] J.D. Parsons and J.G. Gardiner. *Mobile Communications System*. Halsted Press, USA, 1996.

[51] H.V. Poor. Non-Gaussian signal processing problems in multiple-access communications. In *Proc. 1996 USC/CRASP Workshop on Signal and Image Processing*, Ft. George Meade, Maryland USA, 1996.

[52] J.G. Proakis. *Digital Communications, 3rd Ed.* Prentice Hall, New York, USA, 1995.

[53] S.U.H. Qureshi. Adaptive equalization. *Proceedings of IEEE*, 73(9):1349–1387, 1985.

[54] S.U.H. Qureshi and E.E. Newhall. An adaptive receiver for data transmission over time-dispersive channels. *IEEE Transaction on Information Theory*, 19(7):448–457, 1973.

[55] R. Raheli. Per-survivor-processing: A general approach to MLSE in uncertain environment. *IEEE Transaction on Communications*, 43(2/3/4):354–356, 1995.

[56] P.B. Rapajic and B.S. Vucetic. Adaptive decision aided techniques for interference miminization in asynchronous CDMA systems. In *IEEE Int. Symp. on Information Theory*, Trodheim, Norway, 1994.

[57] S.M. Redl, M.K. Weber, and M.W. Oliphant. *An Introduction to GSM.* Artech House, 1995.

[58] M.C. Reed and J. Asenstorfer. A novel variance estimator for Turbo-code decoding. In *International Conference on Communications*, pages 173–178, Melbourne, Australia, April, 1997.

[59] M.J. Riezenman and W. Sweet. Communications. *IEEE Spectrum*, 36(1):29–34, 1999.

[60] H. Robbins and S. Monro. A stochastic approximation method. *Ann. Math. Stat.*, pages 400–407, 1951.

[61] A. Rockstrom and B. Zdebel. A network strategy for survival. *IEEE Communication Magazine*, 36(1):36–40, 1998.

[62] J.P. Ryan. WDM: North america deployment trends. *IEEE Communication Magazine*, 36(2):40–44, 1998.

[63] U. Schoen, A. Hamann, H. Kurzawa, and C. Schmidt. Convergence between public switching and the Internet. *IEEE Communication Magazine*, 36(1):50–65, 1998.

[64] J.M. Senior, M.R. Handley, and M.S. Leeson. Developments in wavelength division multiple access networking. *IEEE Communication Magazine*, 36(12):28–36, 1998.

[65] A.D. Spaulding. Locally optimum and suboptimum detector performance in a non-Gaussian interference environment. *IEEE Transaction on Communications*, 33(6):509–517, 1985.

[66] A.D. Spaulding and D. Middleton. Optimum reception in an impulsive interference environment Part I: Coherent detection. *IEEE Transaction on Communications*, 25(9):910–923, 1977.

[67] A.D. Spaulding and D. Middleton. Optimum reception in an impulsive interference environment Part II: Incoherent detection. *IEEE Transaction on Communications*, 25(9):924–934, 1977.

[68] T.A. Summers and S.G. Wilson. Snr mismatch and online estimation in turbo decoding. *IEEE Transaction on Communications*, 46(4):421–423, 1998.

[69] R.M. Tanner. A recursive approach to low-complexity codes. *IEEE Transaction on Information Theory*, 27(5):533–547, 1996.

[70] G. Ungerboeck. Adaptive maximum-likelihood receiver for carrier-modulated data-transmission systems. *IEEE Transaction on Communications*, 22(5):624–634, 1974.

[71] H.L. Van Trees. *Detection, Estimation and Modulation Theory, Part 1*. MIT Press, Cambridge, MA, USA, 1968.

[72] M.K. Varanasi and B. Aazhang. Multistage detection in asynchronous code-division multiple-access communications. *IEEE Transaction on Communications*, 38(4):509–519, 1990.

[73] M.K. Varanasi and B. Aazhang. Near-optimum detection in synchronous code-division multiple-access systems. *IEEE Transaction on Communications*, 39(3):725–736, 1991.

[74] K.S. Vastola. Threshold detection in narrowband non-Gaussian noise. *IEEE Transaction on Communications*, 32(2):134–139, 1984.

[75] S. Verdu. Minimum probability of error for asynchronous gaussian multiple access channel. *IEEE Transaction on Information Theory*, 32(1):85–96, 1986.

[76] S. Verdu. Optimum multiuser asymptotic efficiency. *IEEE Transaction on Communications*, 34(9):890–897, 1986.

[77] S. Verdu. Adaptive multiuser detection. In *IEEE Int. Symp. on Spread Spectrum Tech. and Applications*, pages 43–50, Oulu, Finland, 1994.

[78] S. Verdu. *Optimal Multi-User Signal Detection.* PhD thesis, University of Illinois at Urbana-Champaign, September, 1984.

[79] A.J. Viterbi. Error bounds for convolutional codes and asymptotically optimum decoding algorithm. *IEEE Transaction on Information Theory*, 13(4):260–269, 1967.

[80] A.J. Viterbi. Furthur results on optimal decdoing of convolutional codes. *IEEE Transaction on Information Theory*, 15(11):732–734, 1969.

[81] A.J. Viterbi. convolutional codes and their performance in communication systems. *IEEE Transaction on Communications*, 19(10):751–772, 1971.

[82] X.D. Wang and H.V. Poor. Robust multiuser detection in non-Gaussian channels. *IEEE Transaction on Signal Processing*, 47(2):289–305, 1999.

[83] L. Wei and Rasmussen L.K. A near ideal noise whitening filter for an asynchronous time varying CDMA system. *IEEE Transaction on Communications*, 44(10):1335–1362, 1996.

[84] L. Wei, Rasmussen L.K., and Wyrwas R. Near optimum tree-search detection scheme for bit-synchronous multiuser CDMA system over Gaussian and two-path fading channels. *IEEE Transaction on Communications*, 45(6):691–701, 1997.

[85] K. Wesolowski. An efficient DFE and ML suboptimal receiver for data transmission over dispersive channels using two-dimensional signal constellation. *IEEE Transaction on Communications*, 35(3):337–339, 1987.

[86] N. Wiberg. *Codes and Decoding on General Graphs.* PhD thesis, University of Linkoping, Sweden, 1996.

[87] Z. Xie, C. Rushforth, R. Short, and T.K. Moon. Joint signal detection and parameter estimation in multiuser communications. *IEEE Transaction on Communications*, 41(7):1208–1215, 1993.

[88] Z. Xie, C.K. Rushforth, and R.T. Short. A family of suboptimum detectors for coherent multiuser communications. *IEEE Journal on Selected Areas in Communications*, pages 683–690, 1990.

[89] Y. Sato Y. Li, B. Vucetic. Optimum soft-output detection for channels with intersymbol interference. *IEEE Transaction on Information Theory*, 41(3):704–713, 1995.